
university of groningen
faculty of science and engineering
computing science
sc@RUG 2023 proceedings
20th SC@RUG
2022-2023
Rein Smedinga, Michael Biehl (editors)
rug.nl/research/bernoulli

SC@RUG 2023 proceedings

Rein Smedinga
Michael Biehl
editors

2023
Groningen

ISBN (e-pub): 978-94-034-3017-1
Publisher: University of Groningen
Title: 20th SC@RUG 2022-2023 proceedings
Computing Science, University of Groningen
NUR-code: 980

About SC@RUG 2023

Introduction

SC@RUG (or student colloquium in full) is a course that master students in computing science follow in the first year of their master study at the University of Groningen.

SC@RUG was organized as a conference for the 20th time in the academic year 2022-2023. Students wrote a paper, participated in the review process and gave a presentation.

SC@RUG is organized by Rein Smedinga and Michael Biehl, both from the Bernoulli institute. Renée Lutke (School of Science and Engineering) helped with improving the presentaiton skills of the students.

Organizational matters

SC@RUG 2023 was organized as follows:

Students were expected to work in teams of two. The student teams could choose between different sets of papers, that were made available through the digital learning environment of the university, *Brightspace*. Each set of papers consisted of about three papers about the same subject (within Computing Science). Some sets of papers contained conflicting opinions. Students were instructed to write a survey paper about the given subject including the different approaches discussed in the papers. They should compare the theory in each of the papers in the set and draw their own conclusions, potentially based on additional research of their own.

After submission of the papers, each student was assigned one paper to review using a standard review form. The staff member who had provided the set of papers was also asked to fill in such a form. Thus, each paper was reviewed three times (twice by peer reviewers and once by the expert reviewer). Each review form was made available to the authors through *Brightspace*.

All papers could be rewritten and resubmitted, also taking into account the comments and suggestions from the reviews. After resubmission each reviewer was asked to re-review the same paper and to conclude whether the paper had improved. Re-reviewers could accept or reject a paper. All accepted papers[1] can be found in these proceedings.

In his lecture about communication in science, Rein Smedinga explained how researchers communicate their findings during conferences by delivering a compelling storyline supported with cleverly designed graphics. Lectures on how to write a paper, on scientific integrity and on the review process were given by Michael Biehl.

Renée Lutke gave tutorials in small groups about presentation techniques and speech skills.

Students were asked to give a short presentation halfway through the period. The aim of this so-called two-minute madness was to advertise the full presentation and at the same time offer the speakers the opportunity to practice speaking in front of an audience. Renée Lutke, Michael Biehl, and Rein Smedinga were present during these presentations.

The final online conference was organized by the students themselves (from each author-pair, one was selected to be part of the organization and the other doing the chairing of one of the presentations). Students organized the conference by setting up the final program, find a sponsor for the breaks, etc. They also found a keynote speaker, **Judith Bachmann** from **IBM** who spoke about *Retail Industry Automation through Data-Driven Insights*.

The organizing students also created a website for this years conference. This years announcements can be found on **https://www.studentcolloquium.nl/2023/**

The overall coordination and administration was taken care of by Rein Smedinga, who also served as the main manager of *Brightspace*.

Students were graded on the writing process, the review process and the 2 minute madness presentation, the presentation during the conference and on their contribution in the organization of this conference.

For the grading of the 2 minute madness presentations we used the assessments of the audience using the application *Poll Everywhere* and also used this application to find the best presentation of the day according to the audience.

For the presentations during the conference we also used *Poll Everywhere* for the assessments of the audiencee (for 50%) and the assessments of Renée Lutke, Michael Biehl and Rein Smedinga (also for 50%). Poll Everywhere again was used to find the best presentation of the day and to ask the audience about their general finding of the sym-

[1] this year, all but one papers were accepted

posium, resulting in the following outcome:

The gradings of the draft and final paper were weighted marks of the review of the corresponding staff member (50%) and the two students reviews (25% each).

The complete conference was also recorded and this recording was published on *Brightspace* for self reflection.

The best 2 miniute madness presentation, the best conference proesentation and the best paper were awarded with a voucher and mentioned in the hall of fame.

Website

Since 2013, there is a website for the conference, see **www.studentcolloquium.nl**. The website contains all previous symposium announcements, all available proceedings and the hall of fame (see next page).

Thanks

We could not have achieved the ambitious goals of this course without the invaluable help of the following expert reviewers:

- Vasilios Andrikopoulos
- Michael Biehl
- Bochra Boughzala
- Andrea Capiluppi
- Dilek Düstegör
- Steffen Frey
- Mostafa Hadadian
- Boris Koldehofe
- Jiri Kosinka
- Fadi Mohsen
- Ayushi Rastogi
- Saad Saleh
- Asadollah Shahbahrami
- Huy Truong
- Fatih Turkmen
- Cara Tursun

and all other staff members who provided topics and sets of papers.

Also, the organizers would like to thank Renée Lutke for helping with the presentation skills and the *Graduate school of Science and Engineering* for making it possible to publish these proceedings and sponsoring the awards for best presentations and best paper for this conference and our symposium sponsor IBM for providing our keynote presentation and the lunch during lunch break.

Rein Smedinga
Michael Biehl

Since the tenth SC@RUG in 2013 we added a new element: the awards for best presentation, best paper and best 2 minute madness.

Best 2 minute madness presentation awards

2023
Germán Calcedo Pérez and Somak Chatterjee
Transferability of Graph Neural Network Generalisation Techniques
and
Shrushti Kaul and Nikhita Prabhakar
Decentralized Federated Learning - Solutions based on Gossip Protocol and Blockchain

2022
David Visscher and Erwin de Haan
A review of networking the cloud datacentre

2021
Niels Bügel and Albert Dijkstra
Mining User Reviews to Determine App Security

2020
Andris Jakubovskis and Hindrik Stegenga
Comparing Reference Architectures for IoT
and
Filipe R. Capela and Antil P. Mathew
An Analysis on Code Smell Detection Tools and Technical Debt

2019
Kareem Al-Saudi and Frank te Nijenhuis
Deep learning for fracture detection in the cervical spine

2018
Marc Babtist and Sebastian Wehkamp
Face Recognition from Low Resolution Images: A Comparative Study

2017
Stephanie Arevalo Arboleda and Ankita Dewan
Unveiling storytelling and visualization of data

2016
Michel Medema and Thomas Hoeksema
Implementing Human-Centered Design in Resource Management Systems

2015
Diederik Greveling and Michael LeKander
Comparing adaptive gradient descent learning rate methods

2014
Arjen Zijlstra and Marc Holterman
Tracking communities in dynamic social networks

2013
Robert Witte and Christiaan Arnoldus
Heterogeneous CPU-GPU task scheduling

Best presentation awards

2023
Germán Calcedo Pérez and Somak Chatterjee
Transferability of Graph Neural Networks leveraging Graph Structures

2022
Luc Pol and Jeroen Lammers
A High-Level Overview of Minimum Graph-Triangulation Approaches

2021
Niels Bügel and Albert Dijkstra
Mining User Reviews to Determine App Security

2020
none, because of corona virus measures no presentations were given

2019
Sjors Mallon and Niels Meima
Dynamic Updates in Distributed Data Pipelines

2018
Tinco Boekestijn and Roel Visser
A comparison of vision-based biometric analysis methods

2017
Siebert Looije and Jos van de Wolfshaar
Stochastic Gradient Optimization: Adam and Eve

2016
Sebastiaan van Loon and Jelle van Wezel
A Comparison of Two Methods for Accumulating Distance Metrics Used in Distance Based Classifiers
and
Michel Medema and Thomas Hoeksema
Providing Guidelines for Human-Centred Design in Resource Management Systems

2015
Diederik Greveling and Michael LeKander
Comparing adaptive gradient descent learning rate methods
and
Johannes Kruiger and Maarten Terpstra
Hooking up forces to produce aesthetically pleasing graph layouts

2014
Diederik Lemkes and Laurence de Jong
Pyschopathology network analysis

2013
Jelle Nauta and Sander Feringa
Image inpainting

Best paper awards

2023
Xiayo Guan and Lonneke Pules
Machine Learning for Leak Detection in Water Networks
and
Eelke Landsaat and Johanna Lipka
Logistic Regression and Linear Discriminant Analysis: A Comparative Overview and an Empirical Time-Complexity Analysis
and
Mike Lucas and Elnur Seyidov
What makes a great software team?

2022
Erblin Ibrahimi and Sven Hofman
State of the Art: Performance Overview of Black-Box Web Application Scanners
and
Willard Verschoore and Gerrit Sijberen Luimstra
Is it Not Yet Time to Swish? Comparing the ReLU and Swish Activation Functions

2021
Ethan Waterink and Stefan Evanghelides
A Review of Image Vectorisation Techniques

2020
Anil P. Mathew and Filipe A.R. Capela
An Analysis on Code Smell Detection Tools
and
Thijs Havinga and Rishabh Sawhney
An Analysis of Neural Network Pruning in Relation to the Lottery Ticket Hypothesis

2019
Wesley Seubring and Derrick Timmerman
A different approach to the selection of an optimal hyperparameter optimisation method

2018
Erik Bijl and Emilio Oldenziel
A comparison of ensemble methods: AdaBoost and random forests

2017
Michiel Straat and Jorrit Oosterhof
Segmentation of blood vessels in retinal fundus images

2016
Ynte Tijsma and Jeroen Brandsma
A Comparison of Context-Aware Power Management Systems

2015
Jasper de Boer and Mathieu Kalksma
Choosing between optical flow algorithms for UAV position change measurement

2014
Lukas de Boer and Jan Veldthuis
A review of seamless image cloning techniques

2013
Harm de Vries and Herbert Kruitbosch
Verification of SAX assumption: time series values are distributed normally

Contents

Interactive Tools in Computer Graphics

Lars Andringa, Bogdan Popescu

Abstract—This paper summarises the field of computer graphics education by creating a centralised list of educational aids and analysing various tools that help a user interact with computer graphics elements for their advantages and disadvantages. All tools were assigned a category, and each tool was analysed for its advantages and disadvantages. As a result, the reader has a starting point to find the right computer graphics education tool for the job they wish to accomplish, with an emphasis on education in computer graphics.

Index Terms—Computer Graphics, education, raytracting, rasterisation.

1 Introduction

Computer graphics started as a niche, specialized field, in which hardware was expensive and graphics-based application programs that were easy to use and cost-effective were few [21]. Then, when personal computers started integrating built-in graphic displays, and bitmap graphics became affordable, graphics-based applications became very popular. As the domain became more accessible, the interest in learning computer graphics concepts and tools also increased. Computer graphics had to be taught using methods such as textbooks, whiteboards, presentation slides, and websites. This can prove to be a challenge, because of [26]:

- Insufficient background, especially inadequate skills in mathematics and programming.
- Difficulties in understanding geometric concepts such as transformations, projections and 3D modelling.
- Difficulties in solving logical problems and making the connection between theory, programming, application and final visual effects.
- Many students are passive learners and do not interact much with peers and teachers.

In order to address these issues, researchers have developed specialised teaching tools for computer graphics, as such skills are often best learnt by experimenting with concepts and interacting with the resulting renderings [25].

Nowadays, computer graphics is rather interactive, with the user manipulating the content, structure, and appearance of objects using input devices, guided by a variety of interactive tools [21]. These tools play an essential role in computer graphics education, making it easier for individuals to understand and experiment with complex concepts such as rasterization and raytracing. Such tools provide a hands-on and engaging experience that helps users understand the principles of computer graphics and improve their skills in this rapidly evolving field. There is a significant amount of software available to teach such skills. The choice of the particular software package however depends on the specific goals and requirements of the teacher.

1.1 State of the art

No papers have been found doing such a broad categorization of computer graphics tools as this review's objectives. Papers such as [24] focus on illustrating potential applications that can be used in learning about 3D animations and 3D printed models, and also, language learning. The researchers showcase Autodesk 3ds Max, Autodesk Maya, Cinema4D, Adobe Animate CC, Blender3D, and AdobeFlash as tools that can be used by teachers in the process of teaching and learning activities in the classroom. [18] compares 2 approaches, Sketch and WIMP, in tasks for modelling 3D objects. The researchers use Teddy – a sketch-based modelling software, and the more traditional WIMP modelling tools Maya & 3DS Max. They did a qualitative and quantitative analysis in order to identify the benefits of both techniques from the users' perspective. The results show that WIMP systems pose a higher number of instructions in doing 3D model creation tasks, but a lower number of instructions in editing tasks. They are also less repetitive than sketch tools.

- *Lars Andringa is a Software Engineering & Distributed Systems master student at the University of Groningen, E-mail: l.s.andringa@student.rug.nl*
- *Bogdan Popescu is a Software Engineering & Distributed Systems master student at the University of Groningen, E-mail: b.popescu@student.rug.nl*

1.2 Research questions

The goal of this research is for the reader to be able to find the appropriate tools for their purposes after reading the paper. It acts as a summarisation of computer graphics tools and a guide for readers to find the right tool. The research questions reflect this by aiming to provide the reader with the necessary information for their decision.

- *RQ1: What tools exist for Computer Graphics education?*
- *RQ2: What categories do the tools belong to?*
- *RQ3: What are the advantages and disadvantages of each tool?*

1.3 Overview

Section 2 shall introduce the methodology used to find the list of tools, categorise them, and describe them. For the results, section 3 introduces the discovered list of tools, section 4 categorises the tools, and section 5 shall provide the descriptions of the tools together with their advantages and disadvantages. Section 6 shall discuss the results, while section 7 provides the conclusions of the research. Section 8 shall then finish the paper discussing future work other researchers may wish to do to expand upon this paper.

2 Methodology

These research questions shall be answered through a 4-step process. These 4 steps are as follows:

1. Gather a comprehensive list of the most important tools available within the field of computer graphics
2. Categorise the list of tools.
3. Provide a short description of each tool.
4. Provide the advantages and disadvantages of each tool.

For the first step, the list must be generated comprehensively and guaranteed to at least catch the most important tools. For this, multiple methods of discovery must be utilised. This research shall use the following methods:

- Automated search (ChatGPT, Google, SmartCat)
- Snowballing
- Expert opinion

For the second step, the list must be categorised. This shall be done through a method derived from Open-Coding [20], where both of us individually categorise the list and then merge the separate definitions together. This reduces researcher bias and guarantees a more objective outcome.

Steps 3 and 4 are done simultaneously for each tool. A concise description is given, in order to make the reader familiar with some of the important features of the tool. The description is followed by a table highlighting the advantages and disadvantages of the specific software. These factors can either make the tool fit for the reader's purposes or make the tool unusable for the given set of tasks it needs to perform.

3 Tools

The discovered tools are listed in Table 1, together with the method of discovery.

Tool	Discovery method
3ds Max [8]	ChatGPT
Adobe Animate [11]	ChatGPT
Adobe Photoshop [12]	Google
Autodesk Maya [13]	ChatGPT
Babylon.js [14]	ChatGPT
Blender [15]	ChatGPT
Gimp [2]	Google
Houdini [3]	Google
Inkscape [4]	ChatGPT
Lightwave 3D [5]	Google
PixiJS [6]	ChatGPT
Rayground [27]	SmartCat
RePiX VR [22]	Expert opinion
Three.js [7]	ChatGPT
Unity [9]	Google
Unreal Engine [10]	Google
Virtual Ray Tracer [19]	Expert opinion

Table 1: Tool list, and their discovery method

The description of each tool can be found in section 5.

These tools were selected for a number of reasons:

- They are visual tools.
- They are interactive.
- They can help to teach computer graphics elements.
- They are relevant to the scope of this study.

The tools discovered using the language model **ChatGPT** [1] were received after using the query "Provide me with a number of interactive software in computer graphics". The results may vary, as the responses by the AI language model are generated. The received list was filtered manually by checking each tool's features and properties, and removing unwanted software, thus mitigating the language model's bias.

We have also included the tools recommended by our expert reviewer, and they will be taking part in the categorisation and ranking of the study. Other tools were selected using a Google Scholar query "Computer graphics educational tools".

4 Categorisation

The selected tools were classified into a number of categories, for a better grouping of similar features that they hold. As a method derived from Open-Coding [20], both researchers have created a set of categories that they considered to create a good division of tools. Then, an agreement was reached by combining the separate sets. The following 4 categories were discovered:

- 3D engine
- 2D engine
- Web rendering
- Educational

The individual categorisation of each tool is then listed in Table 2

Tool	Category R1	Category R2	Final category
3ds Max	3d modelling	3D engine	3D engine
Adobe Animate	2d animation	2D engine	2D engine
Adobe Photoshop	2d modelling	2D engine	2D engine
Autodesk Maya	3d modelling	3D engine	3D engine
Babylon.js	3d modelling	Web rendering	Web rendering
Blender	3d modelling	3D engine	3D engine
Gimp	2d modelling	2D engine	2D engine
Houdini	3d animation	3D Engine	3D Engine
Inkscape	2d modelling	2D engine	2D engine
Lightwave 3D	3d modelling	3D Engine	3D Engine
PixiJS	2d engine	Web rendering	Web rendering
Rayground	3d rendering	Educational	Educational
RePiX VR	VR tool	Educational	Educational
Three.js	3d modelling	Web rendering	Web rendering
Unity	3d engine	3D engine	3D engine
Unreal Engine	3d engine	3D engine	3D engine
Virtual Ray Tracer	3d rendering	Educational	Educational

Table 2: Tool categorisation

A clear distinction between 3D and 2D tools was created, as the features for these types are different enough, in terms of image dimensionality. Furthermore, there is a number of tools that emphasise on an educational level, as the tools selected by the expert opinion, and tools found in research papers. Also, tools that run on a web browser were separated from other 2D or 3D tools, as it is a distinctive feature. The division between modelling and animation tools was dropped, as a significant number of tools that had graphics modelling as a feature, also included animation creation, so the separation was unclear and subjective.

5 Advantages & Disadvantages

The following results show a short description of each tool and its associated advantages and disadvantages. This section is split up into subsections, which each present the tools for their respective category.

5.1 3D engines

5.1.1 3ds Max

3ds Max is a 3D computer graphics software, widely used in the entertainment industry for creating 3D animations, models, visual effects and simulations [23]. It specializes in computer animations, providing main features such as shaders, particle systems, raytracing and global illumination. It also contains a built-in scripting language, MAXScript.

Advantages	Disadvantages
Powerful 3D modeling tools	Not open source
Powerful scripting language	High learning curve
Large community	Intensive resource consumption
High quality output	

Table 3: Advantages & Disadvantages of 3ds Max

5.1.2 Autodesk Maya

Autodesk Maya is a cross-platform 3D modelling and animation tool. It is similar to 3ds Max, although proving significant differences. It provides the necessary tools to create more complex objects, compared to 3ds Max's massive environments and worlds.

Advantages	Disadvantages
Powerful and performant tools	Compatibility issues
Scripting editor	Not open source
Large community	

Table 4: Advantages & Disadvantages of Autodesk Maya

5.1.3 Blender

Blender is a free and open-source 3D creation software that is used for creating 3D models, animations, visual effects, and more. It has a wide range of general features and tools that make it a popular choice for 3D artists, game developers, and animators. Its UI is highly customizable, which can be both an advantage and a disadvantage.

Advantages	Disadvantages
Open-source	Not Industry Standard
Customisable interface	Non-standard user interface for beginners
Large variety of features	Lack of specialized features

Table 5: Advantages & Disadvantages of Blender

5.1.4 Houdini

Houdini is a 3D animation and visual effects software, known for procedural node-based workflow, which enables users to create complex 3D animations and effects using a customizable interface.

Advantages	Disadvantages
Procedural workflow	Node based procedural workflow can be slower than a straight-ahead workflow
Used for Lighting and FX in the industry	Steep learning curve
A free version is available	

Table 6: Advantages & Disadvantages of Houdini

5.1.5 Lightwave 3D

Lightwave 3D is a 3D modelling and rendering tool built for people who are interested in creating 3D objects to render. It is a relatively powerful tool, trying to compete with other modelling tools like Blender. It has however not been updated for a few years, and some of its technologies are relatively old. It is very interesting for people who just want to see how rendering works, without actually writing any code.

Advantages	Disadvantages
No expected programming knowledge	Has not been updated for a while
Fast learning curve	Technologies are outdated
Easily modify models and experiment with lighting	Less flexible than 3D engines that do support code like Unreal Engine

Table 7: Advantages & Disadvantages of Lightwave 3D

5.1.6 Unity

Unity is a broad 3D engine mainly designed for game development. It contains many features, including more modern features like raytracing, has a large community. Many people argue that Unity's graphic fidelity is slightly less than the one of Unreal Engine, its main competitor, however, it tends to be more beginner-friendly. Applications for it are written in C#. While it mainly focuses on game development, it also has its roots in automotive, architecture and film [17].

Advantages	Disadvantages
Beginner-friendly	Graphically better alternatives
Highly customisable	Requires strong hardware
Large Community	Large installation size
Feature-rich	

Table 8: Advantages & Disadvantages of Unity

5.1.7 Unreal Engine

Unreal Engine is a broad 3D engine designed originally for game development but expanded into other 3D applications like architecture, movie scenery and digital twins. It is one of the most popular 3D engines [16]. It mainly shines in being one of the most feature-rich 3D engines, supporting relatively new features like raytracing, machine learning super sampling and virtualised geometry really well. It also supports a blueprint system for visual programming, which aims to reduce the learning curve, which may especially be useful for non-programmers. Though it must be noted that one can also work with it using C++.

Advantages	Disadvantages
Feature-rich	Complex and easily overwhelming
High-quality scenes	Large installation size
Blueprint system for non-programmers	Requires strong hardware
Large Community	

Table 9: Advantages & Disadvantages of Unreal Engine

5.2 2D engines

5.2.1 Adobe Animate

Adobe Animate is a cross-platform computer animation program that specializes in designing vector graphics for multimedia authoring. It provides a range of animation tools such as motion tweening, shape tweening, bone animation, and lip sync that enable users to create complex animations with ease.

Advantages	Disadvantages
Integration with other Adobe software	Steep learning curve
Wide range of assets	Limited 3D support
Cross-platform compatibility (HTML5 Canvas, WebGL, Flash Player)	Compatibility issues with some web browsers

Table 10: Advantages & Disadvantages of Adobe Animate

5.2.2 Adobe Photoshop

Adobe Photoshop is a powerful multi-layered raster graphics editing software, used for editing and manipulating images, and designing graphics for various purposes. It is the most used tool in digital art, also infamously known for image warping applied to human faces[28].

Advantages	Disadvantages
Integration with other Adobe software	Steep learning curve
Extensive raster graphics tools	Limited vector editing capabilities
Customizable user interface	Limited 3D capabilities
Large community	Not open source

Table 11: Advantages & Disadvantages of Adobe Photoshop

5.2.3 Gimp

Gimp is a cross-platform open-source image editing software. It is trying to compete with its more well-known counterpart, Adobe Photoshop. There are significant differences between the 2 tools, such as the price, industry adoption, and the number of plugins available.

Advantages	Disadvantages
Open-source	Not Industry Standard
Lightweight	Steep learning curve
Cross-platform	Fewer tools available

Table 12: Advantages & Disadvantages of Gimp

5.2.4 Inkscape

Inkscape is a free and open-source vector graphics editor used to create vector images, through SVG (Scalable Vector Graphics). It is a free alternative to Adobe Illustrator. Inkscape provides a powerful and flexible set of drawing tools, including bezier and spiro curves, freehand drawing, and advanced path editing.

Advantages	Disadvantages
Scalable Vector graphics	Limited photo editing capabilities
Open-source	Bad print quality
Small learning curve	Performance issues

Table 13: Advantages & Disadvantages of Inkscape

5.3 Web rendering

5.3.1 Babylon.js

Babylon.js is an open-source JavaScript framework for building 3D games and applications that can run in a web browser. It is designed to make it easy to create interactive and immersive 3D experiences on the web without the need for specialized software or plugins. One significant feature is Playground, an IDE where developers can experiment with Babylon.js framework in a browser-based environment.

Advantages	Disadvantages
Performance on new browsers	Limited support for older web browsers (without WebGL)
Users can access applications without installation	Limited functionality compared to other 3D graphics software

Table 14: Advantages & Disadvantages of Babylon.js

5.3.2 PixiJS

A javascript library that allows for the rendering of 2D content. It makes use of 2D WebGL to allow user to render such content. It is known to be lightweight and performant. The rendered objects can be interacted with on the website, just like any other web component.

Advantages	Disadvantages
Users can access applications without installation	No test environment unlike similar tools
Lightweight	Requires knowledge of how to build websites
	No 3D support

Table 15: Advantages & Disadvantages of PixiJS

5.3.3 Three.js

Three.js is a javascript library for implementing rendering pipelines on the web. It allows one to create and display 3D scenes within standard web applications. It currently supports only WebGL, but addons are available for other rendering methods such as CSS3D and SVG. It is mainly known to be a very lightweight library, providing fewer features than alternatives like Babylon.js.

Advantages	Disadvantages
Users can access applications without installation	No test environment unlike similar tools
Lightweight	High learning curve
Well-suited for building rendering demos	Currently only WebGL support, no WebGPU

Table 16: Advantages & Disadvantages of Three.js

5.4 Educational

5.4.1 Rayground

Rayground is a website where users can write small pieces of code to run a raytracing algorithm. The user can implement the core functions used in raytracing however they want, and the website will deal with the whole program around it. This allows for either rapid prototyping of new ideas or allows students to implement small pieces of code to learn how raytracing works.

Advantages	Disadvantages
Rapidly change code and rerun	Only supports raytracing
No installation required	Relatively slow performance
Education-specific	Locked-in code editor

Table 17: Advantages & Disadvantages of Rayground

5.4.2 RePiX VR

RePiX VR is a ”virtual reality tool for computer graphics education, which focuses on the teaching of fundamental concepts of the rendering pipeline and offers researchers the opportunity to study learning in VR by integrating learning analytics” [22]. The tool is specially made

for computer graphics education and displays the results of calculations and lines on the screen to display how the rendering pipeline is performed.

Advantages	Disadvantages
Education-specific	Can only be used in VR
Highly optimised for teaching the rendering pipeline	Relatively few features
Complex installation process	Requires expensive VR hardware

Table 18: Advantages & Disadvantages of RePiX VR

5.4.3 Virtual Ray Tracer

Virtual Ray Tracer is a project written in Unity by Bachelor thesis students of the University of Groningen. It is a tool meant to teach raytracing through visualisation and an interactive demo. Users can specify settings like recursion depth, shadows, and camera position. The tool will then display the result from raytracing and visualises the raytracing process. It is available on multiple platforms, including mobile and web.

Advantages	Disadvantages
Education-specific	Limited in features
Visualises raytracing without any extra work	Relatively slow in performance

Table 19: Advantages & Disadvantages of Virtual Ray Tracer

6 Discussion

Our research aims to gain an understanding of tools that can possibly be used in Computer Graphics to teach students the basic concepts of the discipline. Our research has found 17 different computer graphics-related tools that can be used for computer graphics education. Our research then divided those tools into 4 categories: 3D engines, 2D engines, web rendering tools and educational-specific tools.

We then provide, for each tool, a concise description in order to make the reader familiar with some of the important features of the tools. More importantly, we provide the reader with a set of advantages and disadvantages that may either make the tool fit for the reader's purposes or make the tool unusable for the given set of tasks it needs to perform.

Our results seem to suggest that a lot of tools have a steep learning curve, which is an important point to consider for education. Note that some tools however may be useful to build apps for students, like for example an Unreal Engine setup where the raytracing lines are explicitly shown. This may mean that for some purposes only the teacher needs to understand the tool, as the teacher can use the tool to build apps for the students to experiment with.

The accessibility of the tool is extremely important when using it in a classroom. For a significant number of computer graphics tools, there is a license requirement for each instance. It is an advantage if a tool is open-source so that the funds that would be allocated to buying software can be used in other departments, for further improving the study environment.

Another big difference between the tools is the hardware support. Computer graphics is inherently a computationally expensive field. Furthermore, computer graphics is a field that can contain a lot of specialised tools, like specialised screens (you cannot demonstrate HDR on a non-HDR screen), virtual reality glasses or altered reality devices. This has a big impact on the choice of tools, as one tool requires more computational work than another tool, and therefore the availability of hardware has to be taken into account. Furthermore, some tools explicitly require virtual reality headsets, which are often hard to have access to in large enough quantities.

7 Conclusion

Lately, computer graphics teachers have been looking into ways to improve education in Computer Graphics through visualisation techniques. Our research has provided educational experts with a list of options, their categorisations, a short description, and their advantages an disadvantages.

To answer RQ1, the 17 discovered relevant tools can be found in Table 1.

To answer RQ2, the categorisations of the 17 tools can be found in Table 2.

To answer RQ3, one can investigate Section 5 to see the advantages and disadvantages of each individual tool.

8 Future work

There are many ways in which this study can expand. Firstly, at the moment, the study only focuses on the categorisation of tools and describing them. One expansion could be creating a defined set of metrics with which comparisons can be created between tools and categories respectively. These metrics can be quantified in various means, and the grades for each metric could be given depending on the metric type. For example, a set of objective metrics can receive boolean grades, or in the case of subjective metrics, a fixed grading system should be put in place. These grades could be awarded in a large-scale survey of computer graphics students who get to work with these tools.

In addition, the computer graphics tools can be compared with each other following the grades received in the survey. These comparisons can help emphasise the strong and weak points of each tool. Also, they can be a tiebreaker for computer graphics teachers in choosing the best tool available for their needs.

Following the grades and the categories created, a ranking system can be put in place. These rankings can be extremely helpful in outlining the best tools available from a specific branch, further allowing teachers to select the most useful tool for their field of study.

Other research can also expand upon this research in both width and depth, with width referring to expanding the list of tools being looked at, and depth referring to zooming in on a specific category of tools like education-specific tools and increasing the size of the listed advantages/disadvantages.

Acknowledgements

The authors wish to thank Jiri Kosinka for supporting the paper.

References

[1] ChatGPT. https://openai.com/blog/chatgpt/.
[2] Gimp. https://www.gimp.org/. Accessed: 2023-03-08.
[3] Houdini. https://www.sidefx.com/. Accessed: 2023-03-08.
[4] Inkscape. https://inkscape.org/. Accessed: 2023-03-08.
[5] Lightwave 3d. https://www.lightwave3d.com/. Accessed: 2023-03-08.
[6] Pixijs. https://www.pixijs.com/. Accessed: 2023-03-08.
[7] Three.js. https://threejs.org/. Accessed: 2023-03-08.
[8] 3ds max. https://www.autodesk.com/products/3ds-max/overview. Accessed: 2023-03-08.
[9] Unity. https://unity.com/. Accessed: 2023-03-08.
[10] Unreal engine. https://www.unrealengine.com/en-US/. Accessed: 2023-03-08.
[11] Adobe animate. https://www.adobe.com/products/animate.html. Accessed: 2023-03-08.
[12] Adobe photoshop. https://www.adobe.com/products/photoshop.html. Accessed: 2023-03-08.
[13] Autodesk maya. https://www.autodesk.com/products/maya/overview. Accessed: 2023-03-08.
[14] Babylon.js. https://www.babylonjs.com/. Accessed: 2023-03-08.
[15] Blender. https://www.blender.org/. Accessed: 2023-03-08.
[16] A. Ciekanowska, A. Kiszczak Gliński, and K. Dziedzic. Comparative analysis of unity and unreal engine efficiency in creating virtual exhibitions of 3d scanned models. *Journal of Computer Sciences Institute*, 20:247–253, Sep. 2021.

[17] C. Coutinho. *Unity (R) virtual reality development with VRTK4*. APress, Berlin, Germany, 1 edition, Mar. 2022.

[18] T. L. de Araujo Machado, A. S. Gomes, and M. Walter. A comparison study: Sketch-based interfaces versus wimp interfaces in three dimensional modeling tasks. In *2009 Latin American Web Congress*, pages 29–35. IEEE, 2009.

[19] W. V. de la Houssaije, C. van Wezel, S. Frey, and J. Kosinka. Virtual ray tracer. In *Eurographics 2022-education papers*, pages 45–52. The Eurographics Association, 2022.

[20] U. Flick. *An Introduction to Qualitative Research*. SAGE Publications, 2009.

[21] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, and J. F. Hughes. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996.

[22] B. Heinemann, S. Görzen, U. Schroeder, J. Bourdin, and E. Paquette. Repix vr-learning environment for the rendering pipeline in virtual reality. *Euro-graphics 2022-Education Papers*, 2022.

[23] Z. Lei, H. Taghaddos, S. Han, A. Bouferguène, M. Al-Hussein, and U. Hermann. From autocad to 3ds max: An automated approach for animating heavy lifting studies. *Canadian Journal of Civil Engineering*, 42(3):190–198, 2015.

[24] N. R. Mansor, R. Zakaria, R. A. Rashid, R. M. Arifin, B. H. Abd Rahim, R. Zakaria, and M. T. A. Razak. A review survey on the use computer animation in education. In *IOP Conference Series: Materials Science and Engineering*, volume 917, page 012021. IOP Publishing, 2020.

[25] T. Suselo, B. C. Wünsche, and A. Luxton-Reilly. Technologies and tools to support teaching and learning computer graphics: A literature review. In *Proceedings of the twenty-first australasian computing education conference*, pages 96–105, 2019.

[26] T. Suselo, B. Wünsche, and A. Luxton-Reilly. The journey to improve teaching computer graphics: A systematic review. 12 2017.

[27] A. A. Vasilakis, G. Papaioannou, N. Vitsas, A. Gkaravelis, B. Sousa Santos, and G. Alford. Remote teaching advanced rendering topics using the rayground platform. *IEEE Comput. Graph. Appl.*, 41(5):99–103, Sept. 2021.

[28] S.-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. A. Efros. Detecting photoshopped faces by scripting photoshop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10072–10081, 2019.

An Overview of Explainable Artificial Intelligence

Niek Löke, Hessel van Oordt

Abstract—Explainable artificial intelligence is a fairly new development in the field of AI which aims to provide the users of AI systems, particularly decision support systems, with explanations for how the decisions are reached. This allows users to have increased trust in predictions made by AI systems and the debugging and improving of existing machine learning and deep learning models. In this paper we investigate XAI methods, the types of methods that have been developed (split into ad hoc and post hoc methods) and compare their applications. The methods discussed include existing interpretable models (i.e. decision trees), explanation algorithms for black box models (LIME and SHAP), and newer XAI models that use fuzzy logic. We also discuss the use of XAI in the context of specific applications for intelligent transportation systems and medical imaging, where AI is used to make critical decisions, thus requiring XAI methods to achieve a human level understanding of the reasoning behind these decisions. Our examination of existing research into XAI shows that it is an increasingly necessary advancement for future AI systems, however it should be reviewed and used on a case by case basis.

Index Terms—XAI, interpretability, Decision support systems.

1 Introduction

In recent times, artificial intelligence (AI) has been a standard occurrence in our daily lives. From personalized ads and music recommendations on your favorite platform to self-driving cars, AI is more prevalent than it has ever been. While these AI applications can have great benefits to the way we live, work and interact with each other, they are not without their flaws and limitations. A significant flaw facing AI recently is the lack of transparency and interpretability of these systems.

As the complexity of these AI algorithms grows, they become increasingly hard to understand, even for data scientists themselves. With these algorithms being so interwoven in our society, a way to properly explain and interpret these algorithms is needed. Explainable AI (XAI) is a field that aims to address this need by developing algorithms and methods to provide humans a way of understanding how and why a decision is made by the AI system. This need is especially prevalent in several high-stakes sectors, such as healthcare, finance, and transportation where the consequences of an AI error could be severe or even fatal.

The need for XAI has become increasingly clear in recent years, as AI systems have shown to have biases and make decisions that are hard to explain. Furthermore, AI has also been getting more and more responsibility, either by giving it partial or full control over important systems. For example, for the last decade self-driving cars are in full development, giving the system partly or full control over the car. But how does the AI make decisions when driving the car? And is it safe if you cannot understand its decision making? XAI can help to alleviate some of these concerns.

In this paper we want to explore the current state of research in XAI, how to implement it, the challenges and pitfalls XAI faces and discuss when it should (or shouldn't) be used. We will begin by giving an overview of the state-of-the-art of XAI research, summarizing the latest research in the field. Here we will also talk about what are important metrics when developing XAI systems, such as the need for fairness, accountability and transparency. We will then briefly explain how XAI can be implemented, including possible limitations and benefits for each implementation. Finally we conclude by discussing XAI use cases for intelligent transportation systems and medical imaging and discussing the need for XAI.

2 Methodoloy

Our approach for this paper involves using an initial paper set as a starting point for the literature review and then expanding the search to form the base for the investigation. This paper set includes the papers [4] [6] [9] as well as the meta surveys [8] and [10]. We then use backward snowballing to find further literature that is significant for our investigation. In addition to this we extend our search using targeted keywords of the relevant algorithms, methods and related technologies to identify papers that give further insight into the state of XAI and its current applications. This allows us to collect papers that relate to the target of our research, which has two aspects. Firstly, our search includes papers that provide a more general overview of XAI and describe, evaluate and compare algorithms and methods used to understand AI models. Secondly, we look for instances of specific applications of XAI, how the problem is approached in these situations and the reasoning behind why it is being developed. With this we will investigate the state of the art of the technology as well as the motivation behind the use of XAI in particular scenarios.

3 State-of-the-art

Despite being a relatively new field, XAI has quickly become a focus point within the larger domain of AI and a promising advancement for developing AI systems that can effectively be used for their intended purpose. With XAI being an emerging field, various studies towards investigating and developing XAI methods have been completed. A survey from 2018 [2] identifies the applications for XAI to include topics such as healthcare, finance, legal decisions, military and transportation. However, the overall research into XAI is without unification or standardisation [8], and though reviews on XAI topics have allowed for the identification of a number of research directions, these are still scattered [10] [4]. As a result of this, there is a variety of XAI algorithms and methods[1] used to understand AI systems. However, the capacity to evaluate and compare such algorithms/methods is still limited [8] and it is not yet entirely clear which direction XAI is heading and how and when it can be appropriately used. Löfstrom et al. [8] as well as Saeed and Omlin [10] have conducted meta-surveys to investigate precisely these aspects of the topic. While [10] aims to identify the general challenges and directions for XAI research, [8] focuses on the evaluation of explanation methods and XAI in the context of decision support systems (DSSs). A DSS is a type of system that can use provided data and an existing knowledge base to help with decision making problems ranging from business applications to clinical decision support systems, which support choices made by healthcare professionals [3] [5]. The increasing use of machine learning to create

[1] Details on specific algorithms/methods is provided in section 4

effective DSSs leads to a need for having effective ways to understand how decisions are made by an AI system, which XAI can provide [8].

3.1 Explainability, interpretability and perspectives of XAI

There are generally two ways in which the explainability of AI is achieved. The human level understanding of how the AI arrives at its result is made possible either by using a fully interpretable model (ad hoc), or by using explanation methods to describe how a black box system has reached its decision (post hoc) [8] [10] . The distinction between ad hoc and post hoc methods is made in both of the meta surveys [8] and [10]. However, [8] maintains a focus on identifying a criterion for evaluating and comparing XAI methods. Survey [10] also discuss the interchangeability of the terms "interpretability" and "explainability" in the context of XAI research and suggest the distinction that "[e]xplainability provides insights to a targeted audience to fulfill a need, whereas interpretability is the degree to which the provided insights can make sense for the targeted audience's domain knowledge" [10]. Research on XAI has been categorized into several distinct perspectives [10]:

- **regulatory:** due to the integration of AI with systems that affect legal decision
- **scientific:** to gain further knowledge revealed by AI systems that is otherwise inaccessible due to the use of black box models
- **industrial:** XAI models can be more easily integrated into industries due to less mistrust by users
- **model developmental:** using XAI methods to better understand black box models can lead to improvements for such models. Such systems can then be easier to understand and debug
- **end-user and social:** the development of effective XAI models increases the trust end-users have in the systems they interact with

It is also noted by the authors that this list is not complete, with more possible perspectives within XAI as well as potential overlap between perspectives.

3.2 Evaluation of XAI methods

One meta-survey [8] reviews 15 literature surveys, with the primary focus of understanding how the relatively new XAI methods emerging in the field can be evaluated and compared. Due to the scattered research and lack of standardisation, attributable to the novelty of the field, there is a difficulty in determining the comparative effectiveness of XAI methods. In [8] three primary aspects of XAI are identified, and the authors further provide a number of criteria within each of these for evaluating explanation methods. The criteria presented are collected based on the definitions, due to the criteria having different possible names in the variety of original publications. Firstly, there is the user aspect, in which the criteria are based more on subjective elements. In the user aspect, [8] identify four criteria; trust - which is the user's willingness to use AI decisions as well as their confidence that it is the correct decision, appropriate trust - which is the user's perspective on the AI decisions based on previous interactions and whether these led to correct predictions, bias detection - which is the ability for the user to identify errors, and explanation satisfaction - which is how well the explanation satisfies the needs of the user. The second aspect is the explanation aspect, which involves focusing on the explanations themselves. In this case there are five criteria: fidelity, identity, separability, novelty, and representativeness. The third aspect discussed in [8] is the model aspect, for which they suggest three distinct criteria. These are:

- *performance:* performance essentially evaluates the quality of the predictions, i.e. whether they are accurate, and is therefore closely related to appropriate trust
- *fairness:* the level of fairness of the model depends on whether it has error patterns. A fair system should have no systematic errors that affect the predictions
- *reliability:* reflects to the user appropriate confidence in the system in specific situations

The two meta-surveys provide useful insight into the contexts that XAI can be applied and the various perspectives that should be kept in mind when creating XAI technology. However, the overall state of the art indicates that the field is very much still in an early state and will require further standardization to develop clearer directions of research and ways to evaluate the new methods XAI research introduces [8] [10].

4 XAI Implementation

From our discussion about the fundamentals of XAI we can start to look at how XAI can be implemented. We have divided these methods up into three distinct categories, namely: using already well defined and explainable algorithms, making current black-box models more understandable or creating whole new models with explainability in mind from the start. We will discuss all three of these categories below.

4.1 Understandable algorithms

The easiest way to implement XAI is to use algorithms that are already clear and easy to understand by humans. A popular example of this is a decision tree, the general layout of a decision tree can be found in figure 1. The decision tree can be used for both classification and regression tasks and can be utilized when only conditional control statements are used, i.e. there has to be a clear division per feature. Decision trees classify data points by a step-wise assessment, one node at a time, starting at the root node and ending at a terminal node. This terminal node is called a leaf node and contains the final output of the algorithm. At each node, two possibilities are given (left or right) and a direction is chosen based on some features. As a decision tree makes a clear decisions at each crossing, it is very easy to be interpreted by humans and can therefore be considered a great example of interpretable AI.

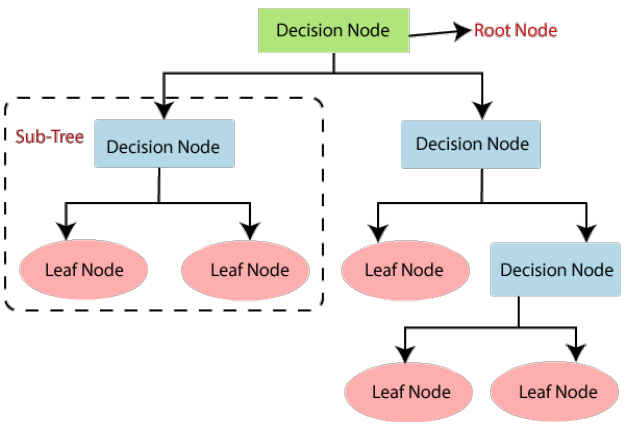


Fig. 1. The general layout of a decision tree. The decision nodes decide what path to take and the leaf node are the final output.

However, decision trees also have some major drawbacks, especially compared to black-box solutions. For one, at each node only two possibilities can be chosen and therefore there are some variable relationships that a decision tree cannot learn. Moreover, as a decision tree grows in size it not only becomes harder to interpret but also becomes time consuming to create. This makes decision trees a cumbersome option when dealing with a large number of features.

4.2 Adapting black-box models

As mentioned, black-box models are systems where we define our inputs and receive our outputs without necessarily knowing how the output was created. In other words, these models are not easily understandable and interpretable by humans. However, the benefits of these black-box models like Deep Neural Networks (DNN), Random forests, and Convolutional Neural Networks (CNN) is that their predictive accuracy can be much higher than their white-box counterparts [12]. A worthwhile endeavour could therefore be to try to make these models more interpretable and transparant.

Two well known techniques to help learn how these models make predictions are LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations). Both of these methods are model agnostic, meaning that they can be applied to any given black-box method we want to use it on. LIME works by perturbing the input data to try and understand how the predictions change. Its output is a list of explanations reflecting the contribution of each feature to the prediction of a data sample. It then uses this to create a simpler, more interpretable model that aims to approximate how the complex model makes its decision [7], called the surrogate model. This surrogate model is therefore more easily understood and can be used to better understand how the original model makes its decision and what features are most important. The general architecture of LIME can be found in figure 2

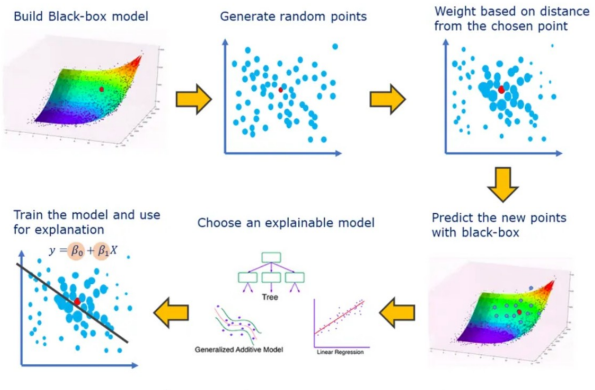


Fig. 2. The general architecture of the LIME method

SHAP's approach is by looking how much each feature influences the output of the model. It does this by calculating the contribution of each feature for a prediction by considering all possible feature combinations and their corresponding Shapley value. The Shapley values are calculated by creating "coalitions" that represent different subset of features and that calculates the contribution of each feature to each coalition. The final feature importance are obtained by taking the average contribution of each feature across all coalitions. This results in a complete overview of how each feature influences the outcome of the model.

Both these methods shed some light on previously unintelligible black-box models and allow us to better understand how these models make their predictions. However, it is important to note that for both these methods it only allows us to better understand what features are important in the models decision making. We still do not know the exact inner working of our models and thus they remain black-box models. We therefore increase the explainability of these models but not to the extend for example a decision tree is interpretable.

4.3 XAI algorithms

A third method of making AI more transparent and explainable is creating algorithms specifically designed for this purpose. A benefit of this approach is that we can make sure that the algorithm really is human-understandable, as this is our aim from the start. This in contrast with black-box (and some white-box) models where first and foremost performance is the goal. An algorithm that was created with explain-ability in mind is fuzzy logic [6].

Fuzzy logic attempts to mimic human thinking, but instead of trying to represent the brains architecture like neural networks, it focuses on how humans think in an approximate rather than a precise way. In other words, it allows for degrees of truth rather than just binary values.

The way it works is that instead of numerical labels it uses imprecise linguistic labels to make predictions. An example for this would be the decision making process when driving; a person wouldn't think "if the car in front is less than 2.5m and the road is 10 percent slippery, reduce the speed by 25 percent". A person would approximate this decision by saying to themselves: the distance to the car ahead is short, and it is slippery so I will slow down. Fuzzy logic can also model these imprecise linguistic concepts by using probabilistic labels as output. An example is given in fig 3, where, for a given income level, it can belong to a low or high income. As we can see, they are well defined for low and high incomes, but take an intermediate value for everything in between. The example shows how an income of 150,000 belongs both to a low and high income, just with differing values, namely 0.7 and 0.3 respectively. This decision can also be seen as more precise as the difference between a dollar more or less can be the difference between a low and high income if binary decision making was used.

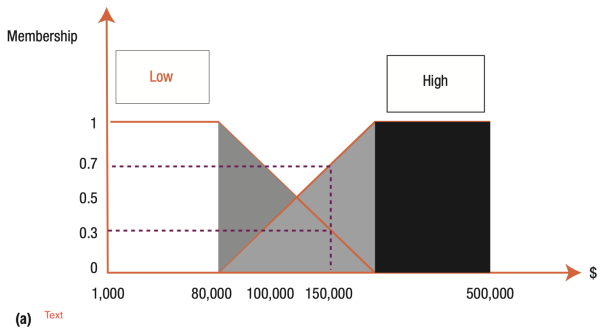


Fig. 3. Fuzzy Logic describing the relation between numerical income and the division into the linguistic labels low and high

We can go one step further by also taking into account the difference between for example various countries or professions. As an example, three different banks were asked their ranges for low income. From this we can construct a type-2 fuzzy set, which can be found in figure 4. This type-2 fuzzy set embeds three separate type-1 fuzzy sets (namely that of bank 1, 2, and 3) and introduces a footprint of uncertainty (FoU) shaded in grey. This FoU allows for additional degrees of freedom that can directly model and handle the uncertainties. In our example we can see that the membership of 150,000 is no longer a crisp value of 0.3 but is now a function ranging from 0.2 to 0.5.

Fuzzy logic has several benefits that allow it to model certain data better than traditional methods. Some of these benefits include its flexibility and granularity that allows it to use uncertain and vague information and output values that are not necessarily binary but have some degrees of truth. It is also very robust and it is already used in several real-world applications such as control systems, decision making and image processing. Finally, as fuzzy logic mimics the way human think using linguistic labels, it is clear to see how humans would be able to understand easily how this model make its decisions, resulting in a model with one of the most important factors in XAI: explainability.

All three of these approaches have their benefits and drawbacks. Fuzzy logic might be the most easily understood method, however it is also less precise. Using decision trees can still be interpretable for

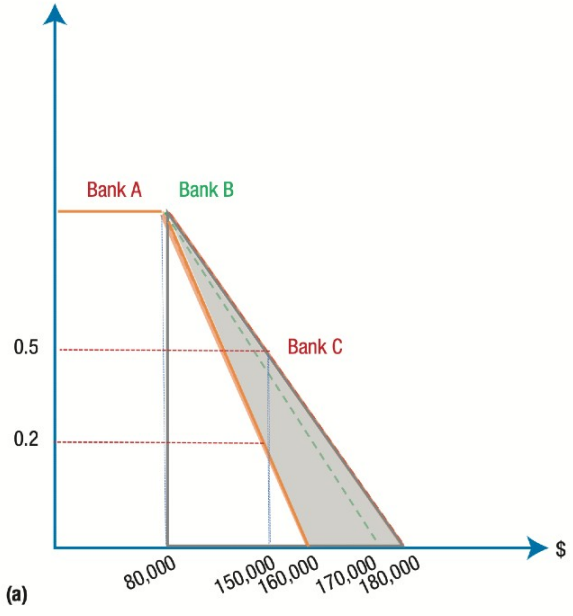


Fig. 4. Type-2 fuzzy logic showing the combination of three different type-1 fuzzy logic models, resulting in an uncertainty range.

small data sets, but falters both for large data sets and in performance compared to black-box models. Finally black-box models give the best performance, but LIME and SHAP only give an overview on how the model uses its features, their precise inner working are still not entirely clear. All of these methods are promising and could be further investigated to try to find the perfect balance between performance and transparency.

5 XAI USE CASES

The question remains, should XAI be used and if so, under what circumstances? As discussed in previous sections, all implementations of XAI have their drawbacks so what method should be used? In this section we will mention several use cases where XAI should be used and also mention some where it should not be used. We will finally discuss these finding in our discussion.

5.1 XAI for intelligent transportation systems

With the integration of AI into DSSs, systems for automated decsions, and further applications for AI, the benefit of XAI methods in such cases is becoming more prevalent. Allowing users to understand the decisions made by AI in such cases increases the trust in these systems and helps prevent errors. Especially in situations where an AI system may be used for critical decisions, potentially involving the lives of multiple people, such as in intelligent transportation systems. Autonomous vehicles, traffic control systems, and other aspects of transportation systems are becoming more integrated with AI systems, which leads to the problem of mistrust when these systems are uninterpretable black boxes. This can be a problem when an error in the system can cause major issues like traffic accidents. XAI offers a solution to such problems by providing the users or communicating systems explanations and an understanding of the reasoning behind the decision [11].

Wollenstein-Betech et al. [11] discuss XAI methods in relation to intelligent transportation systems and present an explanation method for a Traffic Light Control scenario. Their method achieves an explanation of black box models by looking at historical data and generating a representation of the relation between states and actions [11]. Another paper by Mankodiya et al. [9] applies XAI in their investigation for autonomous vehicles in the case of a malicious vehicle attempting to send disruptive or misleading information in a network of communication vehicles. By using a decision tree-based random forest with stacking ML algorithms, they provide a solution to this problem [9] while mainting a human understandable overview of the AI models being used. These papers show the effectiveness that XAI methods can have in achieving explanations for the behaviour of AI. In both cases the methods revolve around the interpretation of black box models, which means these methods can be adaptable for a variety of scenarios. However, this limits the full understanding of how the AI works, and still relies on a complex underlying system that is not entirely human understandable. This is particularly evident in cases where the understanding of the decisions needs to be found from previous data. The alternative, but more complex solution is to make new models that are directly interpretable. This comes with limitations in complexity but would allow for clearer and more reliable explanations on a case by case basis. In both the application to the Traffic Light Control scenario from [11] as well as the use for autonomous vehicles in [9], the use of XAI is not fully integrated into the AI systems themselves, but rather offers insight into past decisions based on historical data, or allows an understanding behind the development of machine learning models. We believe this shows a difficulty that exists when attempting to apply XAI methods in these situations; real time decisions pose an additional challenge for explaining AI systems. Further research on XAI is required to properly integrate the explanation methods with the systems themselves, which can be particularly important in cases where the aspect of individual user interaction with the AI system is significant. In such cases having clear and immediate explanations available to the user increases the trust the user has in the system and can better inform user actions. The user's trust and ability to override the system can be of crucial importance in the mentioned use cases of autonomous vehicles, or instances of (user driven) intelligent traffic light control systems.

5.2 Explainability of medical images

Another paper [1] looks into increasing the explainability of medical images using several currently available XAI techniques. In this study chest X-rays were used in order to diagnose the coronavirus, using black-box CNN models. In order to make this model interpretable the LISA method was used, a combinations of LIME, SHAP and several other methods. The model was able to reach a modest 90% accuracy using a limited dataset. More importantly, due to the implementation of XAI, the researchers were able to identify the most important regions within the image, the regions that had the most influence on its decision. This could allow further research to pinpoint why the wrongly classified samples were classified in this manner. Furthermore, due to the more interpretable CNN, the trustworthiness of the diagnosis will increase, which is essential in the sector of medical diagnosis.

5.3 XAI should not always be used

In our previous segment we have discussed two distinct use cases where XAI brought some tangible benefit to the system in question. Here the transparency of these systems brought more trustworthiness and allowed the systems to be understood more thoroughly. So should XAI be implemented into every AI system? There are several use cases where an algorithm would not benefit or even be negatively impacted by implementing an XAI system. Some examples would be spelling and grammar checks, image recognition systems, online advertising, and music recommendations. For all these examples it would not have much (if any) benefits in understanding their inner workings and either using an explainable algorithm could result in worst results, or explaining a black-box model would lead to unnecessary computations, ultimately slowing down the entire process. As we can see, not every AI system needs to be thoroughly explained and transparent in order to function.

6 DISCUSSION

We have discussed two use cases that benefited from the implementation of XAI, while we also mentioned several use cases where it would

not be beneficial or would even worsen the system by trying to make it explainable. Some sectors where XAI could have a positive impact are: healthcare, financial services, autonomous vehicles and any high stake environment. In these sectors the decisions that are made are important and can have an enormous impact to individuals or society as a whole. Because of this importance a high level of trust is required; when a diagnosis is created you want to know it is the right diagnosis or when an autonomous vehicle is driving you want to make sure that it is doing so safely. The use of XAI to make these decisions more understandable will help with this.

There are also several situations where XAI is not an appropriate measure to use. An example for this would be low-stake decisions, such as the previously mentioned online advertising and music recommendations; an end user would not need to know how these AI systems work. Another example would be time-sensitive decisions, as making AI models more explainable will likely slow down the decision making and this could be detrimental to the systems functioning if XAI would be implemented. An example for this would be emergency medical situations; even though medical situations were mentioned as an example of a good XAI use case, when time is of the essence it would not be wise to use it. In the same vein, resource limited system would also not benefit from XAI, due to the same reason that implementing XAI will cost more time or resources and therefore reduce the performance of the model.

In the end there is no clear answer if XAI should or should not be used. There are several use cases where it would greatly benefit the trust and even performance of a AI system, while in others it would greatly reduce the use-ability of the system. The implementation of XAI should be looked at on a case by case basis, deciding if transparency, explainability and trust is more important or that performance of the system is the most important metric. The former could use an easily explainable algorithm, while the latter could make use of an adapted black-box model to make it more explainable or it could even use no XAI at all.

7 Conclusion

In this paper it was our aim to give a high level overview of the relatively new field of explainable artificial intelligence. We have discussed the state-of-the-art of XAI, where we talked about the current research into the topic, including in what regions its currently being researched/used. We explained what are important metrics when developing XAI, such as performance, fairness and reliability and we also discussed what perspectives are important. Afterwards we went over three distinct methods of XAI implementations, namely using understandable algorithms, adapting black-box model and the creations of new XAI algorithms. In our second to last chapter we introduced several use cases for XAI and also mentions some scenarios where it should not be used. We also introduced the question why, where and if XAI is needed. We concluded with a discussion about these questions and gave our final conclusion on the use of XAI; its use should be looked at on a case by case basis, dependent on the system at hand.

8 Future Work

Our investigation looks at the overall state of XAI, including the various general methods available to achieve explainability for AI systems, the algorithms related to XAI methods, and specific contexts in which XAI can be applied to benefit an existing system. With the relative novelty of XAI, the potential future research can be taken in many directions. We think further investigation into the question of when it is appropriate to apply XAI methods for AI systems can benefit the field. With a clearer insight into the precise advantages and limitations of the distinct XAI methods (interpretable models and explanation methods) it will be more discernible when the use of such methods is appropriate, when the disadvantages of XAI methods may be too limiting to justify their use, and how XAI technology fits into the progress of AI as a whole. Furthermore, we identify the advancement of new, fully interpretable models as an important aspect of XAI, in order to fully integrate XAI technology into AI systems and advance towards real time decisions and explanations.

References

[1] S. H. P. Abeyagunasekera, Y. Perera, K. Chamara, U. Kaushalya, P. Sumathipala, and O. Senaweera. Lisa : Enhance the explainability of medical images unifying current xai techniques. In *IEEE 7th International conference for Convergence in Technology*, pages 1–9, 2022.

[2] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[3] A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, and C. Mooney. Current challenges and future opportunities for xai in machine learning-based clinical decision support systems: A systematic review. *Applied Sciences*, 11(11), 2021.

[4] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[5] A. Bussone, S. Stumpf, and D. O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *International Conference on Healthcare Informatics*, pages 160–169, 2015.

[6] H. Hagras. Toward human-understandable, explainable ai. *Computer*, 51(9):28–36, 2018.

[7] L. Hulstaert. Understanding model predictions with lime. https://towardsdatascience.com/understanding-model-predictions-with-lime-a582fdff3a3b. accessed on 27th of february 2023.

[8] H. Löfström, K. Hammar, and U. Johansson. A meta survey of quality evaluation criteria in explanation methods. In *Intelligent Information Systems*, pages 55–63, Cham, 2022. Springer International Publishing.

[9] H. Mankodiya, M. S. Obaidat, R. Gupta, and S. Tanwar. Xai-av: Explainable artificial intelligence for trust management in autonomous vehicles. In *International Conference on Communications, Computing, Cybersecurity, and Informatics*, pages 1–5, 2021.

[10] W. Saeed and C. Omlin. Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273, 2023.

[11] S. Wollenstein-Betech, C. Muise, C. G. Cassandras, I. C. Paschalidis, and Y. Khazaeni. Explainability of intelligent transportation systems using knowledge compilation: a traffic light controller case. In *IEEE 23rd International Conference on Intelligent Transportation Systems*, pages 1–6, 2020.

[12] Y. Zhang, F. Xu, J. Zou, O. L. Petrosian, and K. V. Krinkin. Xai evaluation: Evaluating black-box model explanations for prediction. In *II International Conference on Neural Networks and Neurotechnologies*, pages 13–16, 2021.

The Privacy and Security Risks of Mobile In-App Browsers

Andrei-Claudiu Veres & Andrei Dumitriu

Abstract— In this study, we investigate the privacy and security risks associated with Mobile In-App Browsers, particularly those implemented with WebView. These integrated browser components allow users to access the web without leaving the host application, providing a seamless user experience. However, our research highlights several risks, such as JavaScript injections, identity confusion, and fingerprinting, that can lead to malicious websites modifying the hosting app, gaining unauthorized access or being able to precisely track users. We also explored how, in some cases, the interfaces of In-App Browsers do not follow industry standards, providing users with less information about the websites they visit and leave them more vulnerable to malicious entities. To address the identified risks, we propose potential solutions, including end-to-end encryption, improved security policies, and the use of VPNs. Furthermore, we provide a short insight into the prevalence of these issues, emphasizing the need for more research into how apps allow users to open links in external browsers. Our paper aims to bridge gaps in the current literature while shedding light on potential mitigation strategies for ensuring safer in-app browsing experiences.

Index Terms—in-app browser, WebView, web browser, user agent string, security, privacy, vulnerability, apps

1 Introduction

The proliferation of smartphones and the apps that run on them has revolutionized the way we interact with technology. With over 86% of the world's population owning a smartphone, mobile applications have become an integral part of our daily lives [27]. While there are a plethora of apps with self-contained functionality, users may sometimes need to access web content linked in the app. Typically, accessing this type of information directs the user to an external browser, which is used to display the content. This situation is not ideal because it disrupts the user experience.

To address this, some apps come with their own in-app browser. In-app browsers allow users to view web content without leaving the app, providing a seamless user experience. However, accessing the web through such browsers also comes with inherent risks to the user's privacy and digital security. In this paper, our focus will be on the implementation of in-app browsers on Android devices, specifically the WebView component. WebView is a Java-based component that enables developers to embed web content in their applications.

Unfortunately, accessing web content via in-app browsers can pose a significant threat to user privacy and digital security. Previous research has indicated that in-app browsers, particularly those utilizing WebView, are vulnerable to a range of security risks and attacks [16][18][26]. In this paper, our goal is to identify how developers and attackers, with malicious intent, take advantage of WebView's implementation to exploit user data. In this context, we will focus on fingerprinting and identity confusion. We also explore potential solutions and practices for mitigating these risks. Moreover, we discuss the trade-off between privacy and usability of in-app browsers, highlighting the importance of considering both factors when designing and implementing them. By analyzing these topics, we aim to provide an overview of the privacy and security risks associated with in-app browsers and inform strategies for addressing them.

In section 2, we begin by presenting the core concepts that pertain to our research. This is followed by section 3, where we discuss vulnerabilities with respect to in-app browsers, specifically those implemented with WebView. We present solutions to mitigate the risks posed by those vulnerabilities in section 4. In the next section, section 5 we will discuss the relation between privacy and usability of in-app browsers. The overview will be summarized in the conclusion, section 6. Lastly, we will end this paper with section 7, where we offer our recommendations for what should be further investigated by the research community.

- *Andrei-Claudiu Veres is with University of Groningen, E-mail: a.veres.1@student.rug.nl.*
- *Andrei Dumitriu is with University of Groningen, E-mail: a.dumitriu@student.rug.nl.*

2 Background

2.1 WebView

WebView is a system component for the Android operating system that enables the display of web content directly inside an app. A developer that wants to add an in-app browser to their app can include the WebView library and create an instance of a WebView class. With this approach, the functionality of viewing web content is added within the app itself, therefore creating a seamless experience for the user [6].

In this paper, we focus on Android apps that offer the functionality of in-app browsers. Therefore, we discuss privacy and security risks that come with the implementation of WebView.

2.2 Super-apps

Super-apps are applications that, with the help of web technologies, present an "app-in-app" structure, delegating some of their functions to sub-apps [29]. An example of such apps is WeChat [29], which in 2020 achieved over 1.225 billion monthly active users and hosts more than 3.8 million sub-apps [1]. This greatly exceeds even the total number of apps available on Google Play: 2.67 million apps [4].

Sub-apps have privileged access to the APIs of their super-app [29]. This allows the sub-apps and super-apps to exchange information. This access is granted through web domains, sub-app IDs as well as capabilities (super-app services and features that are supported in the sub-app) [29].

2.3 Browser fingerprinting

Browser fingerprinting is a method of identifying users based on device information provided by their browser [26]. Fingerprinting does not directly track users, however, this technique is used in the hopes of tracking the user's actions through information about the user's browser and device [19]. The information that is most commonly obtained through fingerprinting includes the user's IP address, time zone, as well as browser and device specifications [19].

One of the ways to obtain this information is through the user agent string. A user agent is any software that retrieves web content and presents it to end users [5]. As such, any web browser is a user agent. As part of any HTTP request, the user agent will send a user agent string. It contains information about the device making the request, such as the type (phone, tablet, personal computer etc.) and the operating system, as well as information about the user agent itself (browser name and version) [12].

More fingerprinting information can be obtained through JavaScript code. For example, a website's JavaScript obtains your device's screen resolution in order to properly display content [19]. JavaScript code can also obtain a user's time zone and IP address.

2.4 Sandbox

A sandbox is a security mechanism that isolates an application, such as a web browser, from the rest of the system. The purpose of this security mechanism is to prevent malicious or unauthorized actions, therefore limiting the impact of potential security vulnerabilities in the application.

In the context of in-app browsers, the sandbox involves running the in-app browser withing a confined environment. The sandbox isolates the browser from the rest of the device's operating system. However, evidence shows that this security mechanism can be maliciously exploited. In subsection 3.2 we show how an attacker take advantage of the WebView implementation by attacking through holes in the sandbox [15].

2.5 Virtual Private Networks (VPNs)

A VPN, or Virtual Private Network, is a technology used to create secure and private connections over the internet. The main uses of this technology are online privacy, bypassing censorship and enhancing online security. A VPN works by encrypting the internet traffic and routing it through a remote server, which is operated by an VPN provider [11].

In the context of this paper, we discuss the relation between VPNs and in-app browsers. In section 4 we discuss how we can use the VPN technology to enhance user privacy.

3 Privacy and Security Risks

3.1 In-App Browsers vs Standalone Web Browser Apps

It should be noted that WebView remains vulnerable to common attacks such as cross-site scripting (XSS), cross-site forgery (CSRF), SQL injection and man-in-the-middle (MITM) attacks [15] [16]. Standalone web apps are vulnerable against these types of attacks too. Naturally, the following question arises, *Is WebView as (in)secure as a standalone web application?*. "Compared with the general browsers, WebView is not secure enough" [28].

There are several reasons for why WebView is more prone to vulnerabilities. Recent research points out that the apps which incorporate in-app browsers do not always adhere to standard browser-specific privacy policies. [26]. Another reason for the lack of security is that in-app browsers do not have the same level of sandboxing and isolation as regular browsers, which facilitate malicious JavaScript injection [18] [28] [15] [16] [20] [24] [22]. "Whereas standalone browsers enforce strong isolation, WebViews can intentionally poke holes in the browser sandbox to provide access to app and device-specific features via a JavaScript interface" [20].

3.2 Threat Models

When it comes to WebView implementations, previous research identified two threats models: **attacks from web pages** and **attacks from malicious applications** [16] [15]. The borrowed figure 1 illustrates the two threat models [16]. On the left side it shows attacks through malicious web pages, while on the left attack through malicious applications.

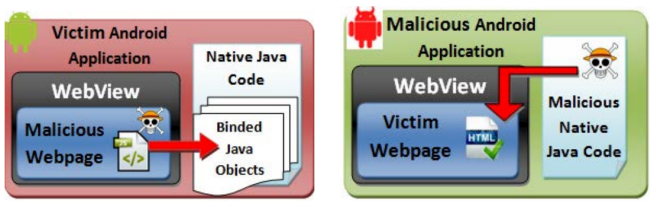


Fig. 1: Threat Models [16].

In the first scenario, attacks from malicious web pages, we assume an application that was designed to serve a web application without ill intent. The goal of the attacker is to compromise the application and their intended web functionality. Such an attack could be accomplished by tricking the victim in loading the attacker's web page into the application. If the aforementioned prerequisites are meet, the malicious hacker could **attack through holes in the sandbox**. For instance, the use of WebView's API *addJavascriptInterface* breaks the browser's sandbox isolation, therefore creating holes in the sandboxes. This circumstance permits the attacker to exploit the *system* as well as the *web application*. [16].

In the second scenario, we assume that the attacker owns a malicious application that is specifically designed to target a web application. For the attack to take place, the attacker needs to lure the user into installing and using their native application for the intended web application. The problem arises because we cannot rely on the security concept of **trusted computing base** (TCB). One way for the attacker to exploit the WebView is through malicious **JavaScript injections**. By using the functionalities provided by WebView, an application can inject its own code into the web page that is loaded in the WebView component. This enables the attacker to manipulate everything in the web page [16]. Another way to exploit the vulnerabilities introduced by the use of WebView is **event sniffing and hijacking**. WebView provides numerous APIs to applications in order to enhance the interaction with the web page. Unfortunately, the attacker can intercept these APIs, hence they can launch sniffing and hijacking attacks which exposes potential sensitive information of the user [16].

Since we established that the implementation of WebView comes with security and privacy risks, we will now examine what can an attacker gain from exploiting those vulnerabilities. In previous research, conducted on 287,000 apps in regard to WebView-related vulnerabilities, it was found that nearly 10% were vulnerable [18]. The research presents a classification of vulnerable application's permissions, therefore showing what an attacker could get access to. In the borrowed figure 2 we can observe that a successful attack could obtain access to SMS and calls functions as well as getting access to system files [18].

Permission (group)	Samples	Percentage of vulnerable samples
RECEIVE_SMS	1,375	4.96%
READ_SMS	1,590	5.73%
WRITE_SMS	933	3.36%
SEND_SMS	1,981	7.14%
SMS permissions	3,124	11.27%
PROCESS_OUTGOING_CALLS	355	1.28%
CALL_PRIVILEGED	134	0.48%
PHONE_CALL	0	0%
Call permissions	382	1.38%
WRITE_EXTERNAL_STORAGE	16,711	60.26%
INSTALL_PACKAGES	1,241	4.48%
Installation permissions	16,727	60.32%
READ_PHONE_STATE	18,935	68.28%
READ_CONTACTS	3,304	11.91%
ACCESS_FINE_LOCATION	11,022	39.75%
ACCESS_COARSE_LOCATION	12,923	46.60%
Privacy permissions	21,197	76.44%

Fig. 2: Permission of Vulnerable Application [18].

3.3 Fingerprinting

Before discussing WebView fingerprinting information, we will first examine how it works in Google Chrome. Google Chrome used to provide the phone's model number as well as its build number in the user agent string [26]. That changed in 2018, when the build number was removed [26]. Additionally, since 2021, Google Chrome provides a privacy sandbox. If enabled, the phone's model number is also removed from the user agent string [26].

However, WebView user strings contain both the model and build numbers. Additionally, they provide a country code for the user's location, as well as their preferred language [26]. As such, users are much more identifiable when using WebView browsers, compared to Chrome with privacy sandboxing, as shown by *Cover Your Tracks* [2], a website that determines how identifiable user agent strings are. Researchers have used this website to compare the uniqueness score of WebView and Google Chrome [26], with the comparison being shown in Table 1.

Browser	Uniqueness (1/X)
WebView	X= 218256
Google Chrome	X = 218112
Google Chrome with privacy sandboxing	X=838.98

Table 1: Fingerprint uniqueness for different browsers.

Additionally, researchers have found 1646 Android apps that come with browsers that use unencrypted HTTP communications to send information such as the device IDs, Google Ads IDs and IP Addresses [26]. This type of communication can be easily intercepted by man-in-middle attacks. These attacks would allow an attacker to obtain this information, in addition to altering the server's response, without the user noticing anything different about the website [26].

What is more, fingerprinting can be enhanced through JavaScript assigning a unique ID to WebView browsers [26]. Studies have shown that Android objects can be changed with JavaScript code [25]. One change that can be made is to transmit unique Ids to Android objects [26]. An example of such code, seen in the literature, is showcased in Figure 3. Such code violates two security policies: modifying the app object, violating its integrity, as well as assigning a unique device Id without asking for permission, which goes against Android Privacy Policies [26].

```
JavaScript:if(window.Application)
{
  Application.setDeviceUid(""APA91bG956w4WPzLIh
  DCHdcnIdbigwApzJzX-WFCkrKRcpJMr9Xw0kbAAxjBYj-
  f6UnVrfeMWRhuPlQIiv8np8733GgHzHm6QHLMeK1
  -InIkhWvxq9yjGb_i2a5WdxIQmaAl-QP3aHHIqK9XTGJiiPpJo
  _dXqkVNzQ"");
}
```

Fig. 3: JavaScript code assigning an ID to Android objects [26].

3.4 Identity Confusion

This security flaw is specific to super-apps. As previously stated, sub-apps use privileged APIs to communicate with their super-apps. However, due to poor design and implementation, super-apps often do not comply with the least principle privilege [29]. This principle states that security architecture should be designed in such a way that each entity has the minimum authorizations and system resources required for it to function [21]. This leads to situations in which an entity can receive unintended privileged API access [29]. This is known as identity confusion [29].

Domain Name Confusion refers to the case of identity confusion where a malicious website passes a web domain verification to access a super-app's API. This happens when the web page displayed in WebView that invokes the privileged API has a different web domain from the domain checked by the super-app [29]. There are two ways to achieve this.

One of those ways is Timing-based Confusion [29], which relies on a race condition. In this case, two different super-app threads handle the API request and the identity check. Between the moments when those threads are started, the malicious web page can change from its actual identity to a privileged identity, thus obtaining unauthorized access [29].

The other type of Domain Name Confusion is Frame-Based Confusion [29]. This type of Confusion relies on an inline frame (also known as iframe), which allows a HTML page to be loaded within a different HTML page. The iframe can act on behalf of the top-frame, using its identity. That is because WebView APIs return only the top-frame's URL [29]. This means that, for example, a malicious advertisement on a sub-app's page can access the privileged APIs of the super-app.

Another variant of Identity Confusion is App ID Confusion, whereby a malicious web page can access a privileged API using a privileged App ID [29]. This can only happen if a malicious web domain is loaded by a sub-app.

The third and final type of Identity Confusion is known as Capability Confusion [29]. This can occur if the super-app makes use of unprotected APIs, with the assumption that only legitimate sub-app developers would have knowledge of them [29]. However, if a sub-app that makes use of such APIs is reverse-engineered, then the APIs can be used by malicious websites. It is worth noting that almost half of super-app runtime APIs are hidden [29]. However, Capability Confusion can also occur through flaws in the APIs of sub-apps. For example, a malicious web page can make a request to a sub-app's API, which then forwards the request to the super-app.

4 Solutions to Mitigate Privacy and Security Risks

Previous research suggest methods to mitigate attacks. A rather "obvious way to thwart traffic tempering is a complete end-to-end encryption" [18]. This approach should mitigate man-in-the-middle (MITM) attacks. Other countermeasures could be, "including origin checks that will drop request that do not match certain IP addressed or are not encoded using a predefined SSL certificate" [18].

Other researchers propose modifications to the WebView security policies in order to enhance the security. They claim that the policies they propose would protect 60% of vulnerable applications with "little burden on developers" [7].

A case study on 465 participants found effective the design of security cues. These cues would alert the user of OAuth-WebView related risks [17].

Notable effort was put in creating systems capable of automating, to an extent, the process of checking apps that incorporate in-app browsers for vulnerabilities. These tools could be used by the user to test the applications of interest before use in order to be aware to what risks are they exposed to [20] [8].

A solution for user tracking, specifically fingerprinting, is the use of virtual private networks (VPN). This approach involves implementing obfuscation techniques and randomization of packet sizes and timing [14]. These techniques can make it more difficult for third parties to analyze the user's internet traffic and reduce the likelihood of pattern recognition.

5 Usability aspects related to security concerns

This section covers how certain aspects of the user experience and interface of WebView In-App Browsers can increase the security risks that users are subject to.

5.1 Displaying Security Information

Depending on each app's implementation of WebView, different In-App Browsers display different information about visited web pages. Some in-app browsers show the domain name, if not the full URL of the page, together with a title, at the top of the view. There are, however, apps that do not display the link, or not even the title of the page [30]. This could be a problem for users if they are directed to a page they believe to be legitimate, but are in actuality malicious websites, posing as legitimate ones. If the user is not shown the URL of web pages, they would be lacking important information they could use to identify such a page. Another security-related usability problem is that some in-app browsers do not properly indicate whether a page uses HTTPS or HTTP, neither showing a corresponding icon nor the scheme portion of the URL, which indicates the transfer protocol [30]. As such, the user can not properly judge the security of the pages they land on when using such browsers. Examples of both WebView In-App Browsers that do and do not display the information described above can be found in the borrowed Figure 4 [30]. Usability problems such as these make it harder for users to avoid the problems discussed

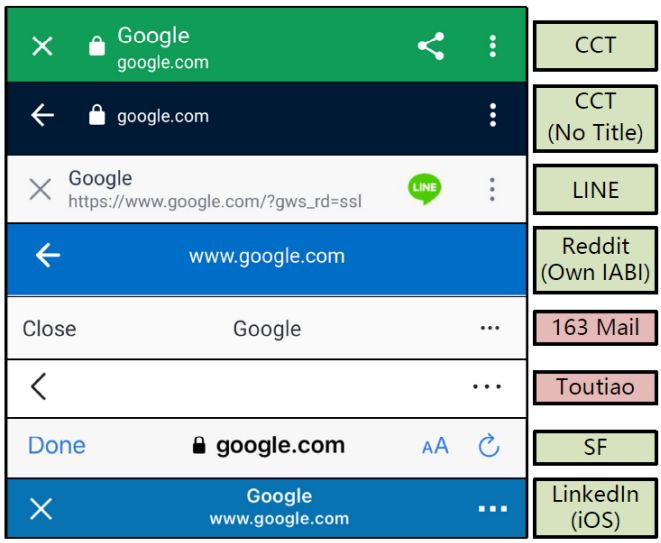


Fig. 4: Examples of proper and improper information display in the top sections of In-App Browsers.

5.2 Browsers Selection Options

Considering the privacy and security risks In-App Browsers pose to their users, we wanted to investigate if and how apps provide users with the option to open their web links in an external browser. To do so, we have used Google Scholar, a large database of scientific papers, to search for information regarding this topic. We used the search query *("webview" OR ("in-app" AND "browser")) AND ("choice" OR "choose" OR "alternative" OR "prompt" OR "select" OR "forced")*.

Unfortunately, little research seems to have been in order to identify trends and changes in the landscape of smartphone apps when it comes to providing users with the option to use an external browser instead of the app's proprietary browser. Looking at the first 2 pages of results, our search in Google Scholar has yielded no research papers that address this topic. However, we can give you an overview of the current state of some popular Android and iOS apps. Table 2 shows the list of apps we have investigated, together with their number of downloads from Google Play.

App	Number of Google Play downloads
Instagram	Over 1 billion
Facebook	Over 5 billion
TikTok	Over 1 billion
Reddit	Over 100 million
Gmail	Over 10 billion
Telegram	Over 1 billion
Twitter	Over 1 billion

Table 2: Investigated apps and their number of downloads from Google Play[3].

We will start with Facebook, which used to provide users with the option to choose their preferred browser (in-app or external) whenever they would open links in the app [23]. That option has been removed in 2021, but later replaced with the ability to change the preferred browser from the app's settings [23]. Reddit, Gmail, Telegram and Twitter provide similar options in their app settings [10]. Instagram does not provide any way to completely avoid using its own browser when opening links in the app, but, once a page has loaded in the Instagram browser, the user can choose to then open it in their default browser [10]. The situation is similar with TikTok's in-app browser, from which you can open the current page in your default browser [13]. The default method for opening links in any of the apps that we mentioned above is in the in-app browser, and the option to change to a different browser is not immediately apparent to all users.

As for why these apps are set up like this, no official reason seems to have been given for the user experience of using an external browser with these apps. One could suggest that the app developers wish to obtain as much data as possible, and the in-app browser is a way to do that. It could also be argued that for some users, the fact that they are not provided with options that lead to an apparently similar result is an improvement to their user experience. However, this is all speculation, and we have no way of providing an official reason.

This topic should, however, be researched more thoroughly. The reason for that is that, as previously discussed, in-app browsers can pose more serious privacy and security risks compared to a regular mobile browser, yet the default option in all of these apps for opening links is the in-app browser. As such, the scientific community should pay more attention to how common these in-app browsers are, why they are used and what option the user has to avoid using them.

6 Discussion and Conclusion

6.1 Threats to Validity

In this subsection, we will enumerate the threats to the validity of our paper.

1. **Limited data:** The number of applications we analyzed in subsection 5.2 is very limited. While the apps that are examined are some of the most downloaded Android applications, the fact that we only researched 5 social media apps, 1 messaging app and 1 email app means that these observations should not be generalized.

2. **Bias:** Personal bias can always influence a research paper. The researchers of this review could have inadvertently emphasised or de-emphasised aspects of their research due to their personal biases. Additionally, the papers referenced here could have suffered from the same type of bias.

6.2 Discussing Advantages and Disadvantages of WebView

One of the main advantages of using an in-app browser is the very fact that the user experience remains contained in one app. If an app requires an external browser for some of its functionality, it is possible for the user to find themselves constantly switching back and forth between that app and the browser app, hurting the user-friendliness [30]. Another advantage is that with in-app browsers, app features could be implemented through web technologies, allowing for easy updates to those features without the user needing to download the update from the App Store or Google Play [9].

However, the WebView implementation of In-App Browsers has a number of problems in terms of privacy and security that are not as prevalent, or in some cases even present, in standalone browser applications. A major security concern is JavaScript injections, which allow an app's browser to make changes to the web pages it visits. Perhaps more concerning, however, is the fact that malicious web pages can make changes to the app that contains the WebView instance. Thus, attackers can gain the app's permissions, gaining access to the phone's contacts and location, as well as the ability to write messages, make calls and install software. Additionally, malicious websites can easily modify the applications which contain browsers in order to assign unique Ids to the device, which facilitates user tracking.

This is compounded by the fact that WebView already sends more information about the device than typical browsers. These fingerprinting techniques, in combination with other tracking technologies, such as cookies, make tracking user much easier in WebView.

All of these risks are increased in super-apps. Our paper clearly showcases that the super-app structure is vulnerable to a much wider range of attacks compared to other types of apps that use WebView.

Unfortunately, our paper shows that the research community does not know enough about how widespread these issues are. A better picture of how apps provide users with the choice between their own browser or an external one would give greater insight into not only how prevalent in-app browsers are within their own apps, but also give an indication to the awareness of users when it comes to in-app browsers.

There are several potential risks associated with the use of in-app browsers. However, it is worth noting that there are also solutions available that can help mitigate these risks to some extent.

Some of the more notable solutions discussed in this paper are: the use of *end-to-end encryption* that mitigates MITM attacks, implementation of better polices that would significantly reduce vulnerabilities and the use of VPNs to avoid fingerprinting.

7 Future Work

As discussed in subsection 5.2, we were unable to find scientific research to help us establish the existence of any trends in how apps offer their user the choice to use an external browser. Such information would be valuable to the research community, given the heightened privacy and security risks of in-app browsers that were highlighted in this paper. Given the lack of insight into how in-app browsers are used in the industry, it is hard to estimate the impact of this technology, as well as how widespread are its risks.

Similarly, further research into the operating system themselves is necessary. To better understand the risks that come with the implementation of in-app browser, we suggest that more effort should be allocated towards studying the default configuration of operating systems in regard to the choice of browser.

Acknowledgements

We would like to thank professor Fadi Mohsen for his input, as well as the student reviewers, for their help in improving this paper.

References

[1] China bytes vol. 1: Wechat, new trends and chinese wisdom. https://www.linkedin.com/pulse/china-bytes-vol-1-wechat-new-trends-chinese-wisdom-camellia-yang/?trk=public_profile_article_view.

[2] Cover your tracks. https://coveryourtracks.eff.org/.

[3] Google play. https://play.google.com/store/apps.

[4] Number of available applications in the google play store from december 2009 to march 2023. https://www.linkedin.com/pulse/china-bytes-vol-1-wechat-new-trends-chinese-wisdom-camellia-yang/?trk=public_profile_article_view.

[5] W3c definition of user agent. https://www.w3.org/WAI/UA/work/wiki/Definition_of_User_Agent.

[6] what is android webview ?

[7] E. Chin and D. Wagner. Bifocals: Analyzing webview vulnerabilities in android applications. In Y. Kim, H. Lee, and A. Perrig, editors, *Information Security Applications*, pages 138–159, Cham, 2014. Springer International Publishing.

[8] M. A. El-Zawawy, E. Losiouk, and M. Conti. Vulnerabilities in android webview objects: Still not the end! *Computers & Security*, 109:102395, 2021.

[9] P. Hazarika, R. R. CP, and S. Tolety. Recommendations for webview based mobile applications on android. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pages 1589–1592, 2014.

[10] J. Hogan. Turn off in-app browser for android apps. https://www.bollyinside.com/articles/how-to-turn-off-in-app-browser-for-android-apps/.

[11] S. Khanvilkar and A. Khokhar. Virtual private networks: an overview with performance evaluation. *IEEE Communications Magazine*, 42(10):146–154, 2004.

[12] J. Kline, P. Barford, A. Cahn, and J. Sommers. On the structure and characteristics of user agent string. In *Proceedings of the 2017 Internet Measurement Conference*, IMC '17, page 184–190, New York, NY, USA, 2017. Association for Computing Machinery.

[13] F. Krause. ios privacy: Announcing inappbrowser.com - see what javascript commands get injected through an in-app browser. https://krausefx.com.

[14] L. Kuypers. Vpn traffic fingerprinting. 2022.

[15] T. Luo. *Attacks and countermeasures for WebView on mobile systems*. PhD thesis, 2014. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2022-11-01.

[16] T. Luo, H. Hao, W. Du, Y. Wang, and H. Yin. Attacks on webview in the android system. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, page 343–352, New York, NY, USA, 2011. Association for Computing Machinery.

[17] F. Mohsen and M. Shehab. Proposing and testing new security cue designs for oauth-webview-embedded mobile applications. In *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*, pages 443–448, 2017.

[18] M. Neugschwandtner, M. Lindorfer, and C. Platzer. A view to a kill: WebView exploitation. In *6th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 13)*, Washington, D.C., Aug. 2013. USENIX Association.

[19] J. Oliver. *Fingerprinting the Mobile Web*. PhD thesis, Master Thesis. London, UK: Imperial College London, 2018.

[20] C. Rizzo, L. Cavallaro, and J. Kinder. Babelview: Evaluating the impact of code injection attacks in mobile webviews. In M. Bailey, T. Holz, M. Stamatogiannakis, and S. Ioannidis, editors, *Research in Attacks, Intrusions, and Defenses*, pages 25–46, Cham, 2018. Springer International Publishing.

[21] R. Ross, V. Pillitteri, K. Dempsey, M. Riddle, and G. Guissanie. Protecting controlled unclassified information in nonfederal systems and organizations. 2020.

[22] F. H. Shezan, S. F. Afroze, and A. Iqbal. Vulnerability detection in recent android apps: An empirical study. In *2017 International Conference on Networking, Systems and Security (NSysS)*, pages 55–63, 2017.

[23] M. Tee. How to force the facebook app to use an external browser to view links. https://www.maketecheasier.com/force-facebook-app-use-external-browser/.

[24] D. R. Thomas, A. R. Beresford, T. Coudray, T. Sutcliffe, and A. Taylor. The lifetime of android api vulnerabilities: Case study on the javascript-to-java interface. In B. Christianson, P. Švenda, V. Matyáš, J. Malcolm, F. Stajano, and J. Anderson, editors, *Security Protocols XXIII*, pages 126–138, Cham, 2015. Springer International Publishing.

[25] A. Tiwari, J. Prakash, S. Groß, and C. Hammer. A large scale analysis of android — web hybridization. *Journal of Systems and Software*, 170:110775, 2020.

[26] A. Tiwari, J. Prakash, A. Rahimov, and C. Hammer. Our fingerprints don't fade from the apps we touch: Fingerprinting the android webview, 2022.

[27] A. Turner. How many smartphones are in the world? https://www.bankmycell.com/blog/how-many-phones-are-in-the-world.

[28] J. Yu and T. Yamauchi. Access control to prevent attacks exploiting vulnerabilities of webview in android os. In *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pages 1628–1633, 2013.

[29] L. Zhang, Z. Zhang, A. Liu, Y. Cao, X. Zhang, Y. Chen, Y. Zhang, G. Yang, and M. Yang. Identity confusion in WebView-based mobile app-in-app ecosystems. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1597–1613, Boston, MA, Aug. 2022. USENIX Association.

[30] Z. Zhang. On the usability (in)security of in-app browsing interfaces in mobile apps. In *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '21, page 386–398, New York, NY, USA, 2021. Association for Computing Machinery.

How Multi-Agent Systems for Anomaly Detection Achieve Decentralization

Rick Timmer and Koen Bolhuis

Abstract—Many different domains benefit from the detection of anomalous events in their operations. However, a system with the goal of detecting anomalies can get quite complicated. By designing a system to use a multi-agent solution, complexities can be abstracted away into separate agents. This allows for the distribution of specific complex tasks as opposed to scaling whole systems. The principle that allows for such a strategy is essentially the decentralization of the actions being taken by the system. Decentralizing a system using a multi-agent approach can be done on multiple levels, ranging from a largely centralized approach to being fully decentralized. In this study, we compared eight multi-agent systems approaches for anomaly detection, and created a spectrum of decentralization achieved by these systems. On the centralized side of the spectrum, we found a system where agents communicate amongst each other as well as with a centralized control system. On the other end of the spectrum, we found systems where agents operate fully autonomously to detect anomalies.

Index Terms—Multi-agent solution, anomaly detection, networked system, decentralization.

1 Introduction

With the rising need for distributed systems, more systems are becoming reliant on networking in order to function correctly. As a result, detecting the occurrence of anomalies in such networked systems is an interesting challenge. Anomaly detection is the practice concerned with distinguishing events that do not conform to expected behavior; such events are then referred to as anomalies [4]. Multi-agent systems (MAS) approaches for anomaly detection are relatively well-researched. Anomalies in networked systems can be broadly separated into two categories: anomalies on the network itself, possibly indicating network intrusion, and anomalies in external entities connected to the network. While the former has seen a large amount of research in its own right [1, 10, 12], in this paper, we focus on the latter category.

1.1 Research goal

We compare multi-agent systems solutions across different domains and create a spectrum of decentralization achieved by these solutions. We do this by examining studies that span the following domains:

- Distributed power grids (Section 3.1);
- Smart buildings and cities (Section 3.2);
- Other domains (Section 3.3).

Especially the first two domains are quite well-developed in the area of MAS for anomaly detection [2, 8, 14, 7, 15, 11], which makes them a logical target for comparison. The third category comprises studies in less-researched domains or domains that do not necessarily align with the former two domains. This allows us to consider a wider range of studies. By comparing results from the different domains, we create an overview of how multi-agent systems can be applied for anomaly detection. The comparison focuses specifically on the level of decentralization, governance, and collaboration between agents.

This paper is structured as follows. Section 2 introduces the reader to the multi-agent systems concept and related work in this area. Following the introduction and background, Section 3 gives an overview of different studies on MAS approaches across different application domains and a comparison between the solutions. Then, Section 4 discusses the findings. Finally, Section 5 concludes the paper, while Section 6 lays out opportunities for future research.

- *Rick Timmer (s4567846) is a Master's student in Computing Science at the University of Groningen, email: r.timmer.9@student.rug.nl.*
- *Koen Bolhuis (s3167895) is a Master's student in Computing Science at the University of Groningen, email: k.bolhuis@student.rug.nl.*

2 Background

To help reason about the multi-agent approaches in different domains, it is useful to first understand the MAS concept, and how it differs from non-MAS approaches. Multi-agent systems are systems consisting of multiple autonomous agents. For our review, we use the definition from Dorri et al. [5]:

> "Agent: An **entity** which is placed in an **environment** and senses different **parameters** that are used to make a decision based on the goal of the entity. The entity performs the necessary **action** on the environment based on this decision.
>
> (...)
>
> While an agent working by itself is capable of taking actions (based on autonomy), the real benefit of agents can only be harnessed when they work collaboratively with other agents. Multiple agents that collaborate to solve a complex task are known as Multi-Agent Systems (MAS)."

This definition allows us to reason about the difference between MAS and non-MAS approaches. A MAS with multiple, separate, agents is able to separate the system into smaller blocks each potentially living in its own environment. Each agent is then responsible for sensing its own parameters and actions. This can help to separate the concerns of each piece of software, which in turn may help scale the system for specific tasks. A non-MAS system would have to group environments together in a single one (e.g. utilities as opposed to gas, water, and electricity). As a consequence scaling the system across one of these environments requires the system to scale as a whole. Moreover, a single system with many different responsibilities might grow in complexity more quickly, requiring more awareness of the large number of parameters required for the whole system to function properly.

2.1 Related work

Individually, multi-agent systems and anomaly detection have seen a plethora of prior studies. For example, a survey by Chandola et al. gathered methods for anomaly detection [4], while Dorri et al. performed a survey on multi-agent systems [5].

A review was written by Labeodan et al. giving an overview of multi-agent systems in the context of renewable and sustainable energy [9]. The review briefly covers the decentralized nature of multi-agent systems, calling it a property of agents. This is inherently true under the definition of multi-agent systems referenced above. However, the

level of decentralization is determined by the particular implementation chosen. Some multi-agent systems may implement a centralized governance system whereas others have agents that govern themselves completely.

Whereas the existing works focus on multi-agent systems in general or applications in a specific domain, we create a cross-domain overview in order to obtain insights into how the MAS concept can be applied for anomaly detection in a more domain-agnostic manner.

3 Multi-agent systems in different domains

We now cover the different domains in which multi-agent systems are applied for anomaly detection. First, we cover systems that support distributed power grids. Then, we consider multi-agent systems in smart buildings and smart city environments. Finally, we examine studies that fall outside of the first two domains.

Note that the figures in this section are accompanied by a color code. These colors correspond to the spectrum found in Section 4, and can be used as a visual aid for the reader.

3.1 MAS for distributed power grids

In the distributed power grid domain, algorithms are applied to address traditional shortcomings in power grid systems. These systems traditionally may use a centralized control structure for the distribution networks. This can deliver good performance in the case of small-scale power systems [2]. However, with the ever-increasing energy demands, in turn increasing requirements, the domain is pushed towards developing more safe and more reliable distributed systems. This has helped push the domain into the adaption of multi-agent systems.

One approach proposed by Mohamed et al. makes use of a centralized control center that is responsible for calculating specific information on anomalies detected by an agent [2]. In this specific use, the agents are all responsible for their own segment on the grid where their task is to identify the anomalies. They also have access to a circuit breaker which is used to protect the circuit. The agents are able to share information with each other to identify the specific location of the fault. After the fault is identified the control center is then used to calculate fault distance. An example of this type of architecture is visualized in Fig. 1. In this case, there clearly is a centralized system where agents are used for separating a specific concern from the main system.

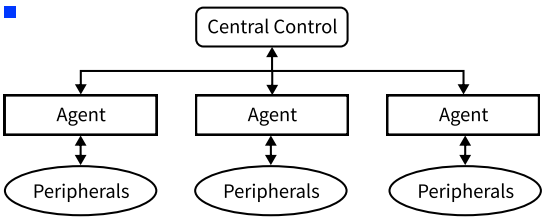


Fig. 1. MAS using a central control system as proposed by Mohamed et al. [2].

Whereas the previously mentioned approach relies on a single type of agent that can communicate with each other, another approach proposed by Shobole is that the agents at that level should not communicate with each other. Instead, they communicate through a coordination agent [14]. Rather than having the relay agents decide themselves whether or not they should operate, they share their information with this centralized agent, which then locates the issue and determines the right action. This approach allows for multiple of these centralized agents to be scaled alongside the relay agents. Another aspect of this approach is that a configuration agent is also put in place, which communicates with the relay agents through the coordination service to inform them which relay should be turned on. Fig. 2 gives an overview of this approach.

A paper by Kiani et al. suggests that by having all neighboring relay agents communicate with each other, they are able to hold a so-called

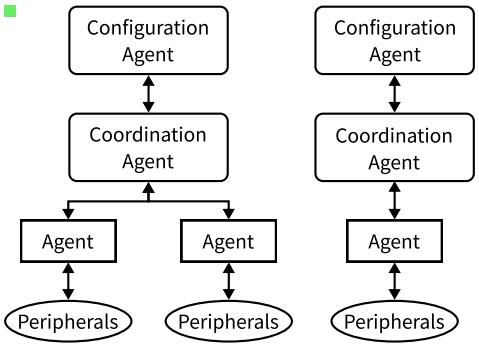


Fig. 2. MAS where agents are responsible for other agents, from Shobole [14].

neighborhood table storing all information of agents in zones around them [8]. By making use of distributed control units, as opposed to a centralized control unit mentioned before, we now have a truly distributed system where the agents are the system as opposed to a part of a centralized system. This has been visualized in Fig. 3. This reduces the significant amount of data that has to be transmitted to the centralized controller. However, since all relay agents do have to communicate with one another, one can imagine that this also creates a lot of network strain. Having the agents split into different neighbor zones, the traffic is limited to a specific zone. However, this would then keep other agents out in the dark at the moment an anomaly is detected. To update agents that should be informed these are grouped as a routed zone, similar to the neighbor zone, which can then be used to broadcast occurrences of faults.

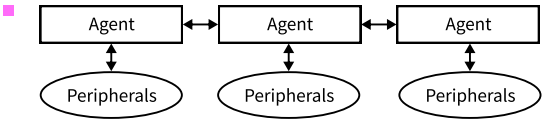


Fig. 3. MAS with no central control system from the paper by Kiani et al. [8].

3.2 MAS for smart buildings and cities

Rapid urbanization has given rise to the concept of smart cities: urban areas integrated and intertwined with intelligent technological infrastructure, where the end goal is improved sustainability from the perspective of governance, citizens, businesses and the environment. [16] This section addresses three aspects of smart cities where anomaly detection plays a role: smart building management, smart campus energy monitoring and parking lot management.

With climate change becoming a more prevalent topic the importance of how we manage our energy is also becoming more widespread. 40% of the energy being used in developed countries is due to buildings, of which 30% is wasted in the United States and Europe [7]. Anomalies in these energy networks are partly to blame for these excessive numbers. One can imagine that with the increase in energy consumption comes an increasing complexity to the management that is required. Existing literature suggests that multi-agent systems can be applied to help identify anomalies and manage the networks overall to result in less wasted energy.

One of these proposed systems is the SANDMAN multi-agent system introduced by Houssin et al. [7], which can identify several types of anomalies in sensor data. A novel part of this system is the self-improving weight system in which each agent is able to determine its weight by cooperating with others. This ensures that the data collected by some agents are seen as more critical by the system when compared to others. The layout of this system is shown in Fig. 4.

With this configuration, the system can adapt its weights without having to rely on a central control unit, making it a more decentral-

ized approach. This calculation of the weights of each agent is done after each data collection cycle, which is every hour for this particular implementation.

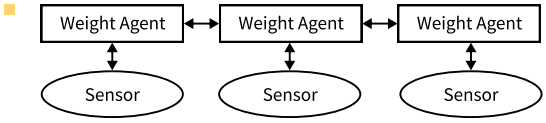


Fig. 4. MAS with agents that determine the weight of individual sensors, from Houssin et al. [7].

Scaling up from individual buildings, energy monitoring systems can also be applied in smart campus environments. A smart campus has the ability to adapt to the needs of various students, due to the fact that the environment is outfitted with different networked devices and sensors. These ubiquitous devices allow inferring, measuring, and understanding environmental indicators [6]. The large amounts of data that are collected in smart campus environments contain a lot of useful information that is waiting to be mined. Anomaly detection on these data streams can, for example, help reduce unnecessary energy waste on the smart campus.

To this end, Weng et al. introduced a multi-agent system for anomaly detection in electricity, water, and natural gas usage on a smart campus [15]. An overview of the architecture in this approach can be found in Fig. 5. In their proposed framework, metering devices continuously collect electricity, gas, and water usage. Then, an agent subsystem, consisting of an electricity agent, gas agent, and water agent, is responsible for the detection of anomalies in the respective consumption patterns. Each agent takes input from metering devices (electricity, gas, and water), which feed into a long short-term memory (LSTM) and autoencoder-based model; each device functions as an input node of the LSTM network. The agents then in turn can send messages to user-facing applications, as well as controlling actuators connected to the agent subsystem.

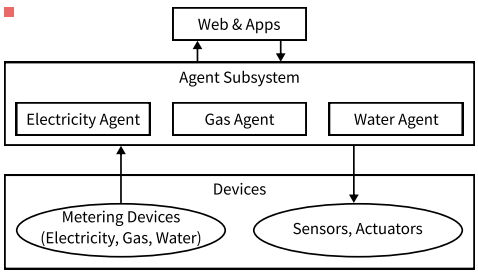


Fig. 5. MAS with independent agents responsible for different utilities, by Weng et al. [15].

Another prevalent aspect of smart cities is parking lot management. More and more parking lots are equipped with security cameras. These cameras provide opportunities to create systems that exploit the information in the camera feeds. This does not only include detection of occupied parking spots, but also extends to anomaly detectiony, e.g. detecting wrongly parked cars or cars that are parked across multiple spots.

A paper by Masmoudi et al. introduced such a system using a multi-agent design [11]. Their architecture is shown in Fig. 6. The system assigns a parking agent (PA) to each parking lot, which is responsible for collecting the parking lot status and transmitting it to end-user applications through a central server.

The PA is also responsible for initializing the position agents (PsA). These agents represent and manage the three-dimensional bounding box of each individual parking spot. During initialization, this bounding box is detected from a camera. After initialization, the PA creates a tracker agent (TA) tasked with the detection of moving vehicles using cameras. When the TA detects a vehicle, it creates a vehicle agent (VA) representing the position and velocity of the vehicle. The VA communicates its position and velocity to each PsA, and receives a confirmation if the VA's vehicle overlaps the PsA's position.

Based on the information the PsA receives from a VA, it can then decide on its degree of occupancy. Moreover, the PsA can communicate with neighboring position agents to determine whether they also detect an overlap with the same VA. If this is the case, the agents use the communicated degrees of occupancy to decide whether a parking anomaly has taken place. Finally, detected anomalies and parking place statuses are communicated back to the PA to update the overall parking lot status in real-time.

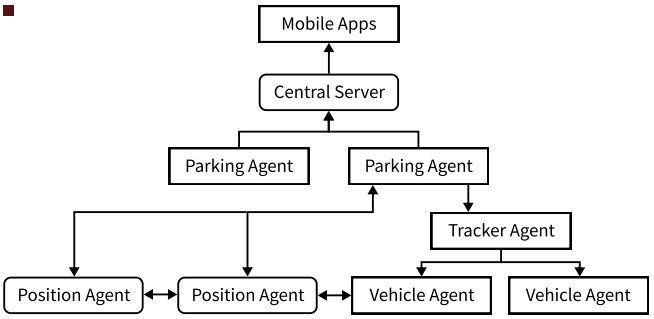


Fig. 6. MAS with many collaborating agents, a central parking agent per parking lot, and a central server for data collection, from Masmoudi et al. [11].

3.3 MAS in other domains

We also found studies outside of the aforementioned domains. The first of these is the system found in the paper by Serino et al. [13]. The multi-agent system in this paper applies anomaly detection to crop fields. Given a historical context, vegetation trends are found with which anomalies can be identified.

The system employs a number of different agents each in charge of a specific task. The data agents are responsible for collecting spectral images and applying general preprocessing. These images are then further processed by the param agents and the contextualizer agents. It should be noted that the contextualizer agent stores its result in a database, which forms the knowledge base of the system. This may influence the way the agents are able to be replicated since the database may have to be scaled alongside it. Lastly, monitoring agents are responsible for detecting anomalies using that data. This agent will receive data from the param agents and query data from the knowledge base made by the contextualizer agents. This means that even though the system is able to scale its data retrieval and preprocessing horizontally, we still need the data to be available on a specific monitoring agent for it to actually be used. A diagram of the system can be seen in Fig. 8.

The final approach we consider is the system designed by Brax et al. [3]. It focuses on anomaly detection in the maritime surveillance domain, a concrete example being the detection of an illegal fishing operation. The system relies on both sensors and a database to provide it with a real-time maritime map of ships in a given area. This is then compared by a Rule Engine which is implemented using maritime regulations.

Anomalies caught by this Rule Engine are then forwarded to their multi-agent system for further detection. Each ship is represented in the system as a ship agent. These agents serve as a connection between the Rule Engine, human operators, and the anomaly detection algorithm. These ship agents also interact with each other to try and spot illicit behavior in collective groups of ships. Every type of anomaly detected by the Rule Engine has its own corresponding anomaly agent in the system. Every anomaly agent is accompanied by three parameter agents which each represent a different parameter. These agents are initially set but are fine-tuned by the feedback of human operators. Fig. 9 gives a visual overview of the described system.

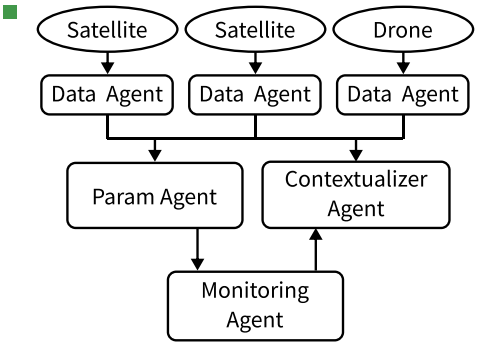


Fig. 8. MAS with independent agents responsible for the processing of image data, by Serino et al. [13].

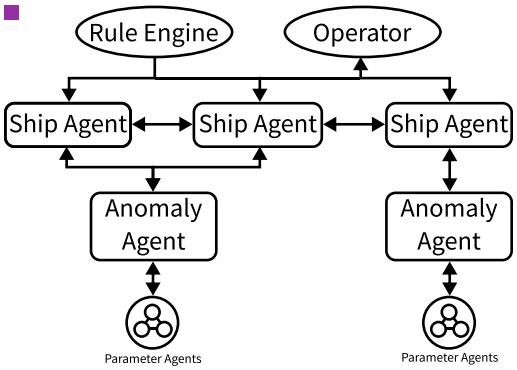


Fig. 9. MAS with independent agents analyzing ship behavior, by Brax et al. [3].

4 Discussion

In the results, we have seen that the systems covered in our study all have different ways in which they achieve a certain degree of decentralization. There does not seem to be a correlation between the domain and the level of decentralization. An example of this are the systems by Mohamed et al. and Kiani et al. [2, 8]. Even though these two systems both belong to the domain of distributed power grids, the former implements a centralized control system whereas the latter uses self-governing agents.

Despite the fact that not every study explicitly mentions the reasoning behind the extent to which they apply different types of agents, we can still draw conclusions. Given a system that is already designed, has been developed, or both, one may want to decentralize certain parts of the system piece-wise. Assuming that certain operations in a system can be done independently of the rest given a set of parameters in a given environment, the systems by Mohamed et al. and Serino et al. [2, 13] are examples of such an approach.

Given a situation in which decentralization plays a larger role in the system's design, the system designer could tend more toward an approach similar to Masmoudi et al. [11]. Their system distributes the level of control on several levels, assigning individual agents to specific, distributed tasks while maintaining centralized control in parking agents and a central server.

Approaches such as the ones suggested by Shobole and Kiani et al. [8, 14] are even more decentralized. A fundamental difference between these two studies is that in the first approach, a group of agents is managed by a corresponding agent that also manages their configuration, while in the second approach the agents manage themselves completely using their neighbors' data as well as their own.

The system by Brax et al. [3] also sees agents cooperating with each other. This is done by having each entity in the system represented by a single agent. In addition, the paper by Houssin et al. [7] uses agents where the neighbors are able to determine their own weight based on the data of their neighboring agents, but without central governance.

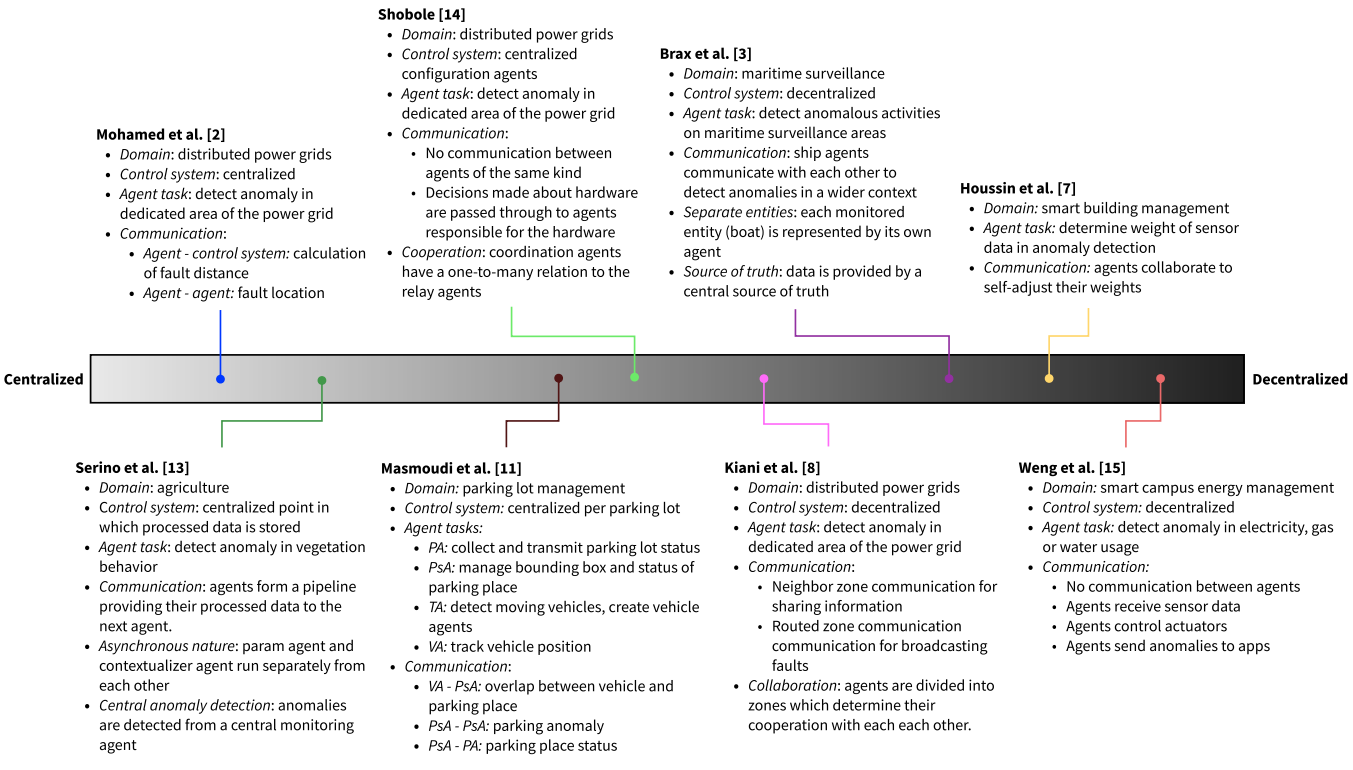


Fig. 7. Decentralization spectrum of multi-agent solutions for anomaly detection in networked systems.

Meanwhile, the framework by Weng et al. considers agents that operate alongside each other, consuming different types of data without necessarily interacting with one another and without any sort of centralized control system [15]. This framework is very general, and while the application domain is the smart campus, such an approach could be applied in many different settings where agents operate on similar types of data streams but which come from different sources.

Finally, the different works used in collecting the results in this paper can be placed on a spectrum where they are ordered from most centralized to most decentralized. We have visualized this spectrum in Fig. 7; we consider the left side more centralized and the right side less centralized. In this figure, we see the system proposed by Mohamed et al. which clearly makes use of a centralized system while still distributing some tasks to the agents [2]. On the opposite side of the spectrum, we placed the system found in the paper by Weng et al. [15]. In this approach, the agents are able to function completely independently to try and complete their actions on a more fine-grained level.

5 Conclusion

In this paper, we discussed multi-agent systems found in research papers which all offer a different level of decentralization based on their design approach. Most of the multi-agent systems that were discussed were quite similar in the level of decentralization achieved, with no fully centralized control systems being in place. It was also noteworthy that the systems used in the distributed power grid domain all diverged in terms of decentralization, having both some of the lowest and highest on the spectrum.

The extent to which agents are able to configure themselves and others also contributes to the level of decentralization obtained. Having the configuration handled by other parts of the systems, whether that be traditional systems or other agents, will impact the system's ability to be distributed. On the other hand, if agents are left to do this for themselves then other decisions have to be made, such as how agents are related to each other on the network. Some ways in which this is handled by the systems we reviewed included having their neighboring agents decided by their physical order, as well as grouping agents via artificial network abstractions, to allow for broadcasting to any interested agent.

6 Future work

Our paper is a qualitative review of existing, mostly theoretical system designs. As such, there is an opportunity to apply the study in a more practical setting. For example, it would be interesting to create or configure real-world implementations of the presented multi-agent systems and compare their effectiveness when applied to the same domain.

Moreover, due to limitations in scope and time, we have not been able to cover all prior literature. Performing a systematic literature review could expand the list of studies as well as cover additional application domains, in order to create a more comprehensive overview of the solution space.

Acknowledgements

The authors would like to thank Dr. Dilek Düştegör for the topic proposal and her helpful guidance during the project.

References

[1] W. Al-Yaseen, Z. ali othman, and M. Z. Ahmad Nazri. Real-time intrusion detection system using multi-agent system. *IAENG International Journal of Computer Science*, 43:1–11, 02 2016.

[2] H. E. M. Azeroual Mohamed, Tijani Lamhamdi and H. E. Markhi. A multi-agent system for fault location and service restoration in power distribution systems. *Multiagent and Grid Systems - An International Journal*, 15:343–358, 2019.

[3] N. Brax, E. Andonoff, and M.-P. Gleizes. A self-adaptive multi-agent system for abnormal behavior detection in maritime surveillance. In G. Jezic, M. Kusek, N.-T. Nguyen, R. J. Howlett, and L. C. Jain, editors, *Agent and Multi-Agent Systems. Technologies and Applications*, pages 174–185, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[5] A. Dorri, S. S. Kanhere, and R. Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6:28573–28593, 2018.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[7] M. Houssin, S. Combettes, M.-P. Gleizes, and B. Lartigue. Sandman: a self-adapted system for anomaly detection in smart buildings data streams. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 14–19, 2020.

[8] A. Kiani, B. Fani, and G. Shahgholian. A multi-agent solution to multi-thread protection of dg-dominated distribution networks. *International Journal of Electrical Power & Energy Systems*, 130:106921, 2021.

[9] T. Labeodan, K. Aduda, G. Boxem, and W. Zeiler. On the application of multi-agent systems in buildings for improved building operations, performance and smart grid interaction – a survey. *Renewable and Sustainable Energy Reviews*, 50:1405–1414, 2015.

[10] F. Louati and F. Ktata. A deep learning-based multi-agent system for intrusion detection. *SN Appl. Sci.*, 2, 2020.

[11] I. Masmoudi, A. Wali, A. Jamoussi, and A. M. Alimi. Multi agent parking lots modelling for anomalies detection while parking. *IET Computer Vision*, 10(5):407–414, 2016.

[12] S. Ouiazzane, M. Addou, and F. Barramou. A multi-agent model for network intrusion detection. In *2019 1st International Conference on Smart Systems and Data Science (ICSSD)*, pages 1–5, 2019.

[13] V. Serino, D. Cavaliere, and S. Senatore. Sensing multi-agent system for anomaly detection on crop fields exploiting the phenological and historical context. In *2021 International Conference on Emerging Techniques in Computational Intelligence (ICETCI)*, pages 43–48, 2021.

[14] A. A. Shobole. Multi-agent systems based adaptive protection for smart distribution network. *Electric Power Components and Systems*, 49(18-19):1432–1444, 2021.

[15] Y. Weng, N. Zhang, and C. Xia. Multi-agent-based unsupervised detection of energy consumption anomalies on smart campus. *IEEE Access*, 7:2169–2178, 2019.

[16] C. Yin, Z. Xiong, H. Chen, J. Wang, D. Cooper, and B. David. A literature survey on smart cities. *Sci. China Inf. Sci.*, 58(10):1–18, 2015.

Transferability of Graph Neural Networks leveraging Graph Structures

Germán Calcedo and Somak Chatterjee

Abstract—Graph Neural Networks (GNNs) are a novel concept that utilises the power of neural networks and deep learning in graph theory. Modern graphs can reach incredibly large sizes and complexities, thus making the job of GNNs much more difficult. To address this, several generalisation techniques have been proposed that aim to mitigate common problems such as over-smoothing or over-fitting of the graph classification predictions. Moreover, the concept of GNN transferability has been recently explored as a pre-training mechanism to utilise features learned from one source graph into multiple target graphs. In this study, we investigate whether the improvements of these generalisation techniques of GNNs can be transferred across multiple graphs. Specifically, we perform this evaluation on graph structures such as k-hop aggregation trees and subgraphs generated by three of the most prominent techniques: node masking, subgraph selection and augmentation-free graph representation learning (AFGRL).

Index Terms—Graph Neural Network, GNN, Transferability, Node Masking, Subgraph, Augmentation-Free Graph Representation Learning, AFGRL.

1 Introduction

Graphs are the optimal way of representing and working with a wide range of datasets, especially when these are comprised of distinguishable entities and clear connections between them. Traditional examples of this are road networks [3] or social networks [14], among others. Nevertheless, novel fields of study also benefit from visualising data as a graph, like protein folding [5], abuse detection [15], computer vision [6] or natural language processing [18].

Recently, several studies have leveraged the power of neural networks in graph theory [11] [12] with effective results, allowing for superior performance when predicting graph classification and labelling by implementing generalisation techniques. This kind of neural network is known as graph neural network (GNN). However, a major issue is increasingly becoming noticeable as the size of the concerning graphs expands continuously: training dedicated and domain-specific GNNs can be costly.

To address this, various techniques have been proposed that help in generalizing GNNs. In this study, we explore Node Masking, proposed by Mishra et al. [16], Subgraph Selection, introduced by Sun et al. [21] and Augmentation-Free Graph Representation Learning (AFGRL) introduced by Lee et al. [13]. There are also explorations into how learning from one graph can be used on an arbitrary target graph. One of the more recent concepts is the use of Transferability [8] [9] [19] as a pre-training procedure.

Our objective is to utilise the concept of GNN transferability analysis as proposed by Zhu et al. [25] and apply it to graphs representations for node masking, subgraph selection and AFGRL with the aim to explore their transferability capabilities. Our contribution is to gain an insight into the maximisation of knowledge transfer when it comes to the different structures such as aggregation trees and subgraphs generated by GNN generalisation techniques. We will be introducing our own experiment in which we compare the transferability for randomly generated graphs and then apply node-masking, subgraph selection and AFGRL on these graphs and see how the transferability is affected.

The paper is organized as follows: Section introduces an outline of GNNs and knowledge transferability and explains our objectives with this paper. Section 2 covers other work related to representation of GNNs. Section 3 explains the concept of analysing transferability of knowledge from one graph to another. Section 4 deals with the different techniques for GNN generalisation and their resulting graph structures. Section 5 outlines the experiment in which compare the transferability of graph structures generated by node-masking, subgraph selection and AFGRL and we discuss the results. Section 6 discusses the final conclusion, as well as the threats to validity and future work.

2 Related Work

Important aspects of unsupervised graph learning have been the representation of GNN and GCN graphs which have been researched extensively [21] along with feature learning of nodes and edges in graphs [7]. The scalability of convoluted neural networks through features such as graph edges has also been explored in terms of how GCNs use first-order approximation of spectral graph convolutions [12] to learn hidden layers of graph features and structures.

There has been much study on unsupervised graph learning on GNNs and recently on how to generalise learning objectives on Graph Neural Networks. The introduction of node masking [16] and ego graph information maximisation is an attempt to generalize the structures of aggregation trees to improve the performance of inductive GNNs.

There have also been studies [25] on how the transferability of a GNN can be maximised through the embedding of graph node features. Methods and frameworks to generalize the properties of graph nodes and edges have also been explored such as in [10] where a framework to pre-train a GNN model using unlabelled data has been used for generating graphs unsupervised.

3 Transferability Analysis

According to Zhu et al. [25] the concept of Transferability Analysis is based on the differences between the training objective's abilities to model a source GNN graph and a graph generated using its knowledge based on the local Laplacians of the two. The paper describes the output of a GNN as a combination of its input node features, its laplacian and the local graph filters. It proposes to improve the graph's utility by learning the filters that are compatible with the other two

- *Germán Calcedo, MSc Computing Science at the University of Groningen. Email: g.calcedo@student.rug.nl*
- *Somak Chatterjee, MSc Computing Science at the University of Groningen. Email: s.chatterjee.6@student.rug.nl*

components for a specific task.

In a direct-transfer setting, a GNN is pre-trained on a source graph unsupervised and is applied on the target graph. The success of the transferability depends on how similar the source and target graphs are in structure and features so that the graph filters are applicable on both graphs. The similarities between graph pairs are measured by graph kernels and the paper introduces the view of a graph as samples of the distribution of its k-hop ego graph features and structures. The encoding of k-hop graph samples from GNN gives concrete structural definitions to graphs which help in measuring the similarities. Since it is not possible for the current GNN training objectives to recover the distribution of ego-graphs, an approach known as Ego-Graph Information Maximisation is proposed by the paper to reconstruct the k-hop ego graph of each node via information maximisation.

3.1 k-hop ego graphs

A k-hop ego graph is a graph in which the k-layer expansion for a node v_i is such that the greatest distance between v_i and any other node in the ego-graph is k.

In a topological space of sub-graphs, a graph G is a viewed as samples of k-hop ego-graphs $\{g_i\}_{i=1}^n$. The structural information of G is stored in the set of k-hop ego graph $\{g_i\}_{i=1}^n$ and their empirical distribution.

3.2 Ego graph information maximisation

The Ego graph information maximisation technique is used to reconstruct distribution information for k-hop ego graphs by reconstructing them for each node. Given a set of graphs $(g_i, x_i)_i$, a GNN encoder Ψ is trained to maximize the amount of mutual information between the graph and the node embedding z_i. Another discriminator, $\mathscr{D}(g_i, z_i) : E(g_i) \times z_i \to \mathbb{R}^+$ I, where $E(g_i)$ is the set of edges, is added to maximize the mutual information by computing the probability that a particular edge e belongs to g_i.

As stated before, the transferability analysis is studying the differences between the ability of the training framework to model the source graph and target graph. The amount of similarities between the source and target graph determines the success of the transferability. In Zhu et al. [25], the transferability of the ego graph was facilitated using the technique of Information Maximization. Figure 1 shows the principle for transfer analysis by comparing a source graph G_0 and two target graphs G_1 and G_2. We aim to use the transferability anal-

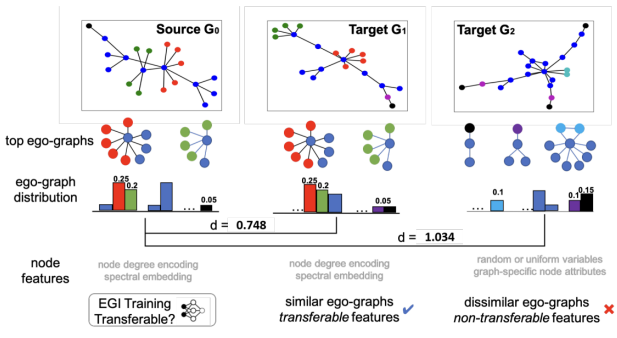


Fig. 1: Transferability analysis framework [25]

ysis to observe the effectiveness of the training models for different GNN models. In our case, we will be examining the similarities in structures and properties of k-hop aggregation trees that undergo node masking in Mishra et al. [16]. The idea is that since node masking prevents repetitive aggregations of a node with the neighbours and restricting the growth factor of a number of nodes. This is done by masking certain nodes in the graph and not including them in the resulting aggregation tree. We can check the similarities in structures for the trees by comparing the compatibility of the graph filters on each tree. If the graph filters are compatible for two aggregation trees, then it can be assumed that the transferability between the two graphs is higher, ie: Transfer learning is very likely to succeed on them.

We will also be using the transferability analysis to understand the effectiveness of subgraph selection in generalizing GNN training. Through the construction of a subgraph neural network and selecting the optimal one to maximise the performance. Using the analysis to compare the structural information between the graphs and finding the similarities between them, we hope to further maximize the accuracy of the subgraph-based GNN.

In essence, transferability analysis hopes to understand the sensitivity of node features to changes in the graph structures by establishing these features as functions of the graph structures and, in the ideal case, to be able to map different features to different structures. This way, we establish the compatibility of graph filters of GNNs to different graphs. Our goal is thus to understand the comparison between the features and structures of the graphs extracted using GNNs that are based on node masking and subgraph selection. We will be exploring what features are respecting of the graph structures in our case.

4 Comparison between graph structures

As transferability analysis measures the ability to utilise GNNs trained on a source graph G_s in an arbitrary target graph G_t by examining the similarity between the structures of both, we now explore the nature of such structures generated by the techniques introduced by Mishra et al. [16], Sun et al. [21] and Lee et al. [13], these being Node Masking, Subgraph Selection and AFGRL respectively.

4.1 Node Masking

Node masking aims to mitigate two issues that arise when working with aggregation-based GNNs: generalisation and scaling. Aggregation-based GNNs compute representation for a node by iteratively combining the representations for its neighbours. This course of action enables aggregation-based GNNs to capture and extract knowledge regarding the structure of the concerning graph effectively, as long as the conditions are right, as exposed in [23] and [24].

Understanding and picturing how aggregation-based GNNs work can be challenging. Thankfully, we can easily represent and visualise them by utilising specific types of trees, as introduced in multiple studies under different names like *tree walks* [2] or *subtree patterns* [20]. We will, however, refer to these trees as *aggregation trees*, for consistency reasons with our concerning study regarding node masking [16].

4.1.1 Aggregation Trees

Aggregation trees are constructed for each node of a graph through the concept of *k-hops*. A *k-hop* can be defined as the set of unique paths that can be reached traversing k edges, starting from any given node. Given a graph $G = \{V, E\}$ where V is the set of vertices and E is the set of edges, we then define an aggregation tree starting from a node v. $v \in V$ as T_v^k, where k is the index of the *k-hop* used for the such tree.

It is important to remark that there are multiple ways to construct an aggregation tree. To enforce that only a unique tree can be defined for each combination of v and k, we require the resulting tree to follow two rules:

- For any given node in the tree, only one path can exist that connects it to the root.
- The tree must have the lowest possible number of nodes.

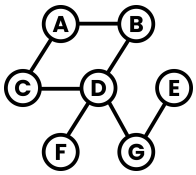


Fig. 2: A sample graph X

As an example, let us consider the graph in figure 2. If we compute successive *k-hops* for the node C we obtain the aggregation trees depicted in figure 3. In essence, aggregation trees represent the structure

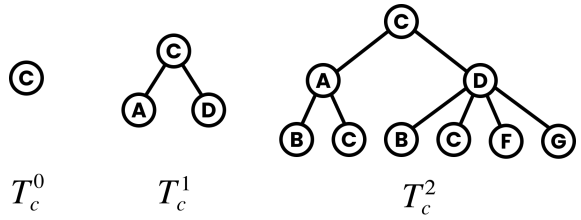


Fig. 3: Aggregation trees obtained from the graph X in figure 2

captured by an aggregation-based GNN, where the number of layers of such GNN is equivalent to the biggest *k-hop* index of the corresponding aggregation tree. This is proved by Mishra et al. [16] through their *Theorem 3.4.*

4.1.2 Addressing generalisation and scaling

Let us now return to the concept of node masking. As we have mentioned before, the basic mechanism of action of aggregation-based GNNs carries two main downsides.

First, as *k-hops*, and in turn aggregation trees, are constructed based on previous iterations, interactions between neighbouring nodes are amplified, thus affecting the ability of the GNN to generalise beyond such structures. This is especially relevant for the root node, as it gets aggregated with itself extensively. In figure 3, T_c^2 shows the aggregation of the root node C with itself on both branches of the tree.

Second, it is not difficult to observe that the number of nodes of aggregation trees grows rapidly as k increases. This unbounded growth results in the inability to effectively train large GNNs once a certain threshold has been reached. This threshold is tightly related to both the computational power of the training platform and the factor by which the number of nodes grows when computing larger *k-hops*.

Node masking addresses both of these issues by minimising the repetitive aggregation of a node with its neighbours and by decreasing the growth factor of the number of nodes. This is achieved by "masking out" specific nodes from the original graph and thus discarding them from being included in the expansion of any aggregation tree. Going back to our running example found in figure 2, suppose two cases. In the first one, we mask node D from the graph X. In the second one, we mask node B. Figures 4 and 5 depict these masked graphs. If we now compute T_c^2 in both cases, we observe that the

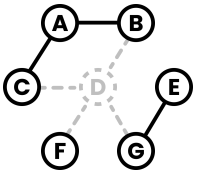


Fig. 4: Node D is masked

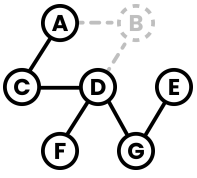


Fig. 5: Node B is masked

resulting aggregation trees differ between themselves and between the original one in figure 3. Figures 6 and 7 represent both aggregation trees. Examining now the obtained aggregation trees we can observe

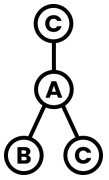


Fig. 6: T_c^2 with D masked

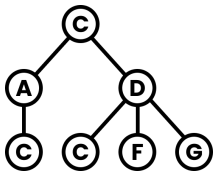


Fig. 7: T_c^2 with B masked

that the amplification of the relations of a node with its neighbours is minimised, as such neighbours are altered by the masking. We can also conclude that the factor of growth of the total number of nodes of the aggregation trees is reduced, as masking out nodes ultimately leads to a lesser number of nodes being expanded. Thus, node masking effectively addresses both the generalisation and the scaling issues that arise from the course of action of aggregation-based GNNs.

We use the collection of structures generated by node masking, i.e. the masked graphs and the resulting aggregation trees to perform transferability analysis, with the objective of determining if node masking improves the transferability for GNNs.

4.2 Subgraph Selection

Similarly to node masking, subgraph selection attempts to tackle a series of issues regarding GNN training, mainly discrimination and interpretability. This is, recalling the generalisation problem described previously, the tendency of GNNs to over-smooth the resulting graph representations, thus rendering the specific features of graphs indistinguishable or to over-fit, augmenting certain structures and resulting in a lack of sufficient interpretability.

Fundamentally, subgraph selection computes graph classifications based on subgraph representations. By obtaining the classification in this way, the subgraph-based GNN preserves graph properties and structures, as this classification is directly dependent on the set of elected sub-graphs. Therefore, the challenge of this approach, and the performance of the subgraph-based GNNs are closely linked to the optimal selection of subgraphs. Although several works utilise a heterogeneous range of sub-structures like motifs [4] or edges [7], we explore only GNNs with subgraph [21]. We refer to the desired optimal graphs that maximise the performance of a subgraph-based GNN as *striking subgraphs*.

To efficiently extract striking subgraphs, Sub et al. propose a novel approach comprised of three distinct modules [21].

4.2.1 Subgraph Neural Network

Initially, we construct a neural network that aims to reconstruct a sketch of the source graph through the combination of its striking subgraphs. To better grasp the inner workings of this technique, let us suppose we want to perform striking subgraph extraction on the graph Y of figure 8. To build this neural network, we first sample a set of

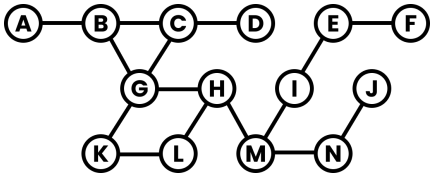


Fig. 8: Another sample graph Y

subgraphs from our source graph. The sampling is parameterised by

two values that act as upper bounds: n limits the number of subgraphs that we sample and s restricts the amount of nodes in each of these sampled subgraphs.

Suppose we perform subgraph extraction with $n = 5$ and $s = 5$. We now sort all nodes by their degree, i.e. the number of edges connected to each node, in descending order. For Y, we have the following ordering: G, B, C, H, ... , A, D, F, J. We now simply take the n first nodes of this ordering and perform breadth-first search until we reach s nodes for each subgraph. Figure 9 depicts the five sampled graphs. The centre node for each of them is coloured blue. A GNN-based encoder is learned to acquire and encode node

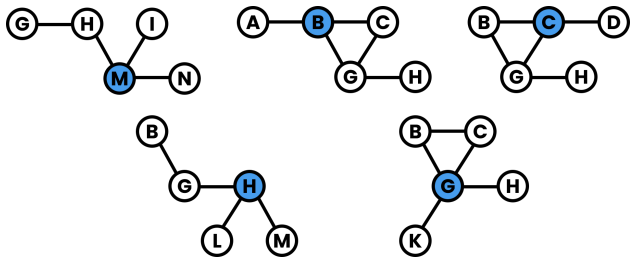


Fig. 9: Sampled subgraphs from Y

representations within the sampled subgraphs. This is the core paradigm of this approach. These subgraph and node representations are later used to compute the original source graph classification without losing the interpretability of its properties and structures.

From the sampled subgraphs, we select those with prominent patterns. This is achieved through a reinforcement pooling module, covered in section 4.2.2, that favors subgraph encodings and representations that are largely retained when projected onto the complete set of subgraph features. In other words, we select the set of subgraphs that best sketches and covers the source graph.

Going back to our running example on graph Y, suppose we end up selecting two subgraphs. There are clear differences on which pair of graphs we select and how they sketch the original source graph. The subgraphs centered around B and C share almost all of their nodes, thus not covering the source graph effectively. However, subgraphs centered around B and M only share two nodes, in turn sketching the source graph Y with higher fidelity. Figures 10 and 11 show this difference. We now reconstruct a sketch of

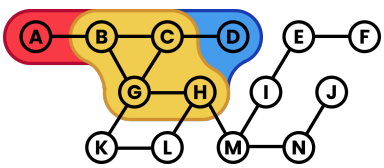


Fig. 10: Section of Y covered by B and C centered subgraphs. Covered by B in red, C in blue and both in yellow.

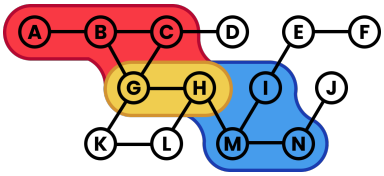


Fig. 11: Section of Y covered by B and M centered subgraphs. Covered by B in red, M in blue and both in yellow.

the source graph by combining its striking subgraphs obtained in the previous step. We denote this sketch with the label ^{ske}, in our example, $Y^{ske} = \{V^{ske}, E^{ske}\}$, where V^{ske} is the set of sketched vertices and E^{ske} is the set of sketched edges. It is important to remark that V^{ske} and E^{ske} do not refer to the vertices and edges of the source graph. Instead, sketched vertices represent complete striking subgraphs alongside their encodings, and sketched edges the connections between them.

The set of striking subgraphs V^{ske} is always included entirely in the resulting sketched graph. Nevertheless, the set of connections E^{ske} is computed based on the connectivity between striking subgraphs. We measure connectivity as the number of common nodes, i.e. overlap in the source graph, between two subgraphs. A threshold b_{com} is defined to only include connections that meet the such threshold in terms of connectivity. This is, any given pair of striking subgraphs must meet b_{com} to have a connection in the sketched graph. Therefore, E^{ske} is defined as follows

$$E^{ske} = \{e_{i,j}\}, \forall \mid V(g_i) \cap V(g_j) \mid > b_{com}$$

where g_i and g_j are any two distinct elements from the complete set of striking graphs $\{g_1, g_2, g_3, \ldots, g_n\}$ and $e_{i,j}$ is a connection between the two.

Finally, we use the methods laid out by Veličković et al. [22] to learn an attention mechanism that computes the embeddings for each of the striking subgraphs. Basically, the attention coefficient for each subgraph is determined by the mutual influence among subgraphs that share its features. These embeddings are converted to label predictions via a softmax function. The probability distributions for each of the predictions are then combined. The index class with the highest probability is determined to be the predicted label of the source graph.

4.2.2 Reinforcement Pooling Module

As we have stated before, the main challenge in Subgraph Selection is optimising and constructing the set of subgraphs to include in the subgraph neural network model. To address this, a reinforcement learning pooling module is leveraged to select the best-performing subgraphs.

To achieve this, the reinforcement pooling module receives reward signals from the subgraph neural network as predictions are computed during training. Such a process tunes this module to select subgraphs that capture important features from the source graph. This carries a major advantage: the selection process is independent of the size and structure of the input subgraphs, thus improving the generalisation capabilities of Subgraph Selection.

4.2.3 Self-Supervised Mutual Information Module

To measure the expressive ability of the representations computed by the subgraph neural network, we utilise the concept of mutual information. Mutual information is, in essence, the quantification of the mutual dependency between two variables. In our specific case, our variables are the representations of the local striking subgraphs and the global source graph.

We seek to maximise the mutual information between both representations. To do so, the estimator proposed by Nowozin et al. [17] is exploited, which takes pairs of subgraph-graph embeddings as input and computes whether they both belong to the same graph. As a quick overview, figure 12 illustrates how all three modules interact with each other to perform Subgraph Selection. We take the set of striking subgraphs as the collection of structures to perform transferability analysis.

4.3 Augmentation-Free GRL (AFGRL)

AFGRL is a self-supervised learning technique introduced by Lee et al. [13] to create alternate views of a graph by taking into account

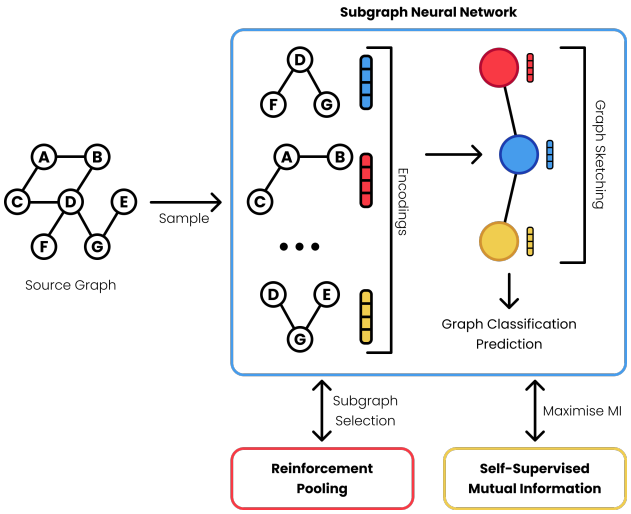


Fig. 12: Subgraph Selection architecture, inspired by [21]

the relational-inductive bias of its structural data as well as its global semantics. This is done by only considering nodes that can be considered as positive samples of node representation, i.e.: nodes which share similar representations in the node embedding space. The node embedding is performed by an online and target encoder, f_θ and f_ξ respectively, that uses the adjacency (A) and feature (X) matrices of the original graph as inputs. The encoders then compute the online and target representations of the graph, denoted by $\mathbf{H}^\theta = f_\theta(X,A)$ and $\mathbf{H}^\xi = f_\xi(X,A)$. For a given query node $v_i \in \mathcal{V}$ in graph $\mathcal{G}$, the cosine similarity is computed between the other nodes by:

$$sim(v_i, v_j) = \frac{h_i^\xi . h_j^\theta}{\| h_i^\xi \| \| h_j^\theta \|}, \forall v_j \in \mathcal{V}$$

Given this knowledge about the representation similarity, the idea is to search for k-nearest-neighbours for each query node v_i and denote them as a set B_i. The idea is that one can expect that the nearest neighbours in the representation belong to the same semantic class as the query node. Figure 13 shows the workflow of the framework. Using

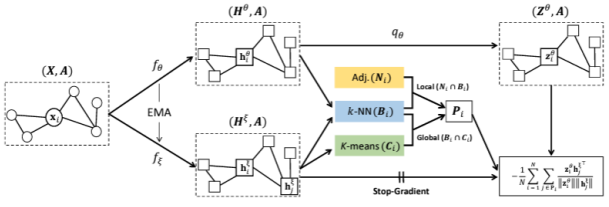


Fig. 13: AFGRL framework [13]

a predictor q_θ, $\mathbf{H}_\theta$ is projected to a predicted embedding Z_θ which is used to compute the total loss along with $\mathbf{H}_\xi$
Although the initial set of positive samples can serve as a reasonable candidate for v_i, there are still problems with it, namely:

- 1. B_i does not take advantage of label information as the set contains samples which are not semantically related to the query node.
- 2. Only considering the nearest neighbours in the representation space neglects structural information inherent in the graph and the global semantics of the graph

To address these problems, the framework introduces a mechanism to filter out false positives and capture the local structural information and the global semantics of the graph is introduced.

4.3.1 Capturing Local Structural Information

In order to filter out false positives from the k-neighbours search, the local structural information among the nodes is used. The structural information is presented in the form of an adjacency matrix which presents the relative inductive bias. This means that for the given node, its adjacent nodes N_i tend to share the same label. The local structural information is denoted as the intersection of the sample set of nodes and the adjacent nodes of the query node, $B_i \cap N_i$. This set of positives is called the *local positives* of the query node.

4.3.2 Capturing Global Semantics

In order to capture the global semantics, a clustering technique is used with the intention of discovering non-adjacent nodes with similar semantic information as that of the query node. This discovery is done by applying a K-Means clustering algorithm to the target representation $\mathbf{H}^\xi$ to organize and cluster nodes into a set of K clusters. This is denoted by $\mathbf{G} = \{G_1, G_2, ...G_K\}$ and the cluster assignment of h_i^ξ, ie: $v_i \in G_{c(h_i^\xi)}$ is denoted by $c(h_i^\xi) \in \{1....K\}$.

The next step is to consider the set of nodes that belong to the same cluster as v_i as the one that is semantically similar globally. The set of nodes that belong in the same cluster as v_i is denoted by $C_i = \{v_i \mid v_j \in G_{c(h_i^\xi)}\}$ The final step is to obtain the intersection between the nearest neighbours and the semantically similar set and denote this set, $B_i \cap C_i$ as the set of *global positives*. What this implies is that nodes that are the nearest neighbours of v_i and belong in the same cluster are considered globally positive neighbours.

It is important to note that the clustering algorithm is sensitive to centroid initialization. Therefore, it is important to perform the algorithm multiple times to ensure robustness of the clusters. In essence, the K-means algorithm is performed M times to obtain M sets of clusters.

The goal is to obtain a set P_i of *real positives* which is a union of the sets of local and global positives. This is denoted by

$$P_i = (B_i \cap N_i) \cup (B_i \cap C_i)$$

Wrapping up, the graph structures that we will use from AFGRL for the transferability analysis are the query nodes, the *real positives* and the edges between them.

5 Experiments

Having explored the generalisation techniques that we are concerned about, these being node masking, subgraph selection and AFGRL, we now lay out the experiments that we use to test their transferability capabilities.

We perform transferability analysis with a source graph against a series of target graphs, first on the source graph itself and then on a set of possible graph structures generated through the techniques.

We would like to make a special remark in that the implementation of such techniques exceeds our current level of understanding and the time span of this study. However, we do have a grasp on the nature of the generated structures and we can create a usable sample accordingly.

As our dataset, we use a widely utilised graph collection: the Facebook social circles provided by Stanford University.[1].

5.1 Our implementation of transferability analysis

Following the guidelines proposed by Zhu et al. [25], we now describe the methodology we have followed to implement our own version of the transferability analysis, levering the k-hop ego graphs of a structure, i.e. its degree distribution.

Suppose we aim to measure the transferability between two graph structures. We denote this measurement by the symbol T_G^H, where both G and H are graph structures. T_G^H is the result of the transferability analysis between G and H.

First, we compute the degree for each node in both graphs, i.e., the number of connections of each node. We then construct the probability distribution PD of each possible degree across each graph. For any given degree k, its corresponding probability p_k is computed as follows

$$p_k = N_k / N$$

where N is the total number of nodes in the structure and N_k is the number of nodes with k degree.

From this point onwards, we will refer to the probability distributions of graphs G and H as PD_G and PD_H respectively. We now compute the weighted average probability distribution PD_W between PD_G and PD_H. Both graphs are assumed to have the same weight.

$$PD_W = \frac{1}{2}(PD_G + PD_H)$$

The transferability analysis introduced by Zhu et al. [25] leverages the Jensen-Shannon Divergence between PD_G and PD_H, which is bounded between 0 and 1. Values close to 1 indicate a high divergence between both graph structures, thus suggesting low transferability. On the contrary, values close to 0 indicate low divergence and high transferability. The Jensen-Shannon Divergence JSD between two probability distributions P and Q is computed as follows

$$JSD(PD_P \parallel PD_Q) = \frac{1}{2}(KL(PD_P \parallel PD_W)) + \frac{1}{2}(KL(PD_Q \parallel PD_W))$$

where PD_W is the weighted average probability distribution between P and Q and KL is the Kullback-Leibler Divergence between the distribution of one of the analysed graphs and the weighted probability distribution. The Kullback-Leibler Divergence is a numerical value that can be obtained in the following way

$$KL(P \parallel Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

where P and Q are probability distributions and X is the sample space on which they are defined.

Our final measurement of transferability is determined by

$$T_G^H = JSD(PD_G \parallel PD_H)$$

5.2 Results

We now present the results of performing the transferability analysis between a set of target graphs and two source graphs. We compare four structures of the source graph.

- **Base**: the source graph as is.
- **NM**: graph structures derived from **Base** with node masking.
- **SS**: graph structures derived from **Base** with subgraph selection.
- **AF**: graph structures derived from **Base** with augmentation-free learning.

Graphs are denoted by $F_n(N,E)$, where n is the name of the graph in the dataset, N is the number of nodes of such graph and E the number of edges. In blue, values that outperform the graph as is. In red, graphs structures that perform worse. In bold, the best transferability achieved for each target graph.

We first present the results of taking $F_{107}(1034,53498)$ as our source graph.

Target Graph	Base	NM	SS	AF
$F_0(333,5038)$	0.3072	0.2835	0.2933	**0.2767**
$F_{348}(224,6384)$	**0.1840**	0.1952	0.1867	0.1923
$F_{414}(150,3386)$	0.2997	0.2669	0.2567	**0.2493**
$F_{686}(158,3312)$	0.2593	**0.2443**	0.2489	0.2476
$F_{698}(61,540)$	0.5416	0.5188	**0.5151**	0.5202
$F_{1684}(786,28048)$	0.1107	0.1090	0.1113	**0.1077**
$F_{1912}(747,60050)$	0.1623	0.1704	0.1617	**0.1599**
$F_{3980}(52,292)$	0.6316	0.6190	**0.6174**	0.6233

Table 1: $T_{F_{107}}^H$ values, where H is a target graph

Finally, these are the results of taking $F_{3437}(534,9626)$ as the source graph.

Target Graph	Base	NM	SS	AF
$F_0(333,5038)$	0.0935	0.0920	**0.0889**	0.0895
$F_{348}(224,6384)$	0.1474	0.1952	**0.1456**	0.1678
$F_{414}(150,3386)$	0.1432	0.1426	0.1478	**0.1387**
$F_{686}(158,3312)$	**0.1034**	0.1138	0.1089	0.1055
$F_{698}(61,540)$	0.2529	0.2426	0.2433	**0.2412**
$F_{1684}(786,28048)$	**0.1635**	0.1709	0.1670	0.1692
$F_{1912}(747,60050)$	0.3752	0.3760	0.3685	**0.3622**
$F_{3980}(52,292)$	0.3957	**0.3355**	0.3890	0.3840

Table 2: $T_{F_{3437}}^H$ values, where H is a target graph

6 Discussion

From the results depicted in tables 1 and 2 we can conclude that transferability is generally improved when testing it on the explored graph structures.

The techniques tend to perform better when the source graph has a larger number of nodes and edges. This can be expected, as a larger graph can contain more diverse and heterogeneous structures on which to train a GNN. This diversity gets reflected in the set of graph structures that are generated through the different techniques.

Finally, augmentation-free learning yields the best results when it comes to transferability, with node masking and subgraph selection performing similarly, but slightly worse.

6.1 Threats to validity

Possible threats to validity regarding this study encompass different aspects. First, the risk of not having properly implemented the procedure to perform the transferability analysis. Second, the selection of the dataset and the source and target graphs within it may not be optimal or might have included some undesired bias or skew in the results. Lastly, the sample of graph structures we have generated for each of the techniques may not accurately represent the real set of structures that is obtained when performing the actual techniques and not just a rough sampling.

6.2 Future work

As this is a novel field of study, there are various interesting paths to continue this study forward. More datasets can be explored to make the results more robust. The generation of the sample graph structures can be refined or, if within reach, complete implementation of the algorithms and techniques proposed in the papers can certainly reduce the risks of utilising a non-optimal sample. As novel techniques are proposed, transferability analysis can be performed on them to compare the results with previous experiments.

ACKNOWLEDGEMENTS

The authors of this study wish to thank Huy Truong for his guidance throughout the project and the lecturing team from the Computing Science Student Colloquium course at the University of Groningen for providing valuable insight on redacting this study.

REFERENCES

[1] Stanford University, facebook social circle dataset. `https://snap.stanford.edu/data/ego-Facebook.html`.

[2] F. R. Bach. Graph kernels between point clouds. In *Proceedings of the 25th international conference on Machine learning*, pages 25–32, 2008.

[3] R. Bader, J. Dees, R. Geisberger, and P. Sanders. Alternative route graphs in road networks. In *Theory and Practice of Algorithms in (Computer) Systems: First International ICST Conference, TAPAS 2011, Rome, Italy, April 18-20, 2011. Proceedings*, pages 21–32. Springer, 2011.

[4] K. Baskerville and M. Paczuski. Subgraph ensembles and motif discovery using an alternative heuristic for graph isomorphism. *Physical Review E*, 74(5):051903, 2006.

[5] K.-C. Chou. Applications of graph theory to enzyme kinetics and protein folding kinetics: Steady and non-steady-state systems. *Biophysical chemistry*, 35(1):1–24, 1990.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004.

[7] L. Gong and Q. Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9211–9219, 2019.

[8] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[9] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867, 2020.

[10] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1857–1867, New York, NY, USA, 2020. Association for Computing Machinery.

[11] N. Keriven and G. Peyré. Universal invariant and equivariant graph neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[13] N. Lee, J. Lee, and C. Park. Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7372–7380, 2022.

[14] A. Majeed and I. Rauf. Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions*, 5(1):10, 2020.

[15] P. Mishra, M. Del Tredici, H. Yannakoudakis, and E. Shutova. Abusive language detection with graph convolutional networks. *arXiv preprint arXiv:1904.04073*, 2019.

[16] P. Mishra, A. Piktus, G. Goossen, and F. Silvestri. Node masking: Making graph neural networks generalize and scale better. *arXiv preprint arXiv:2001.07524*, 2020.

[17] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

[18] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072, 2018.

[19] L. Ruiz, L. Chamon, and A. Ribeiro. Graphon neural networks and the transferability of graph neural networks. *Advances in Neural Information Processing Systems*, 33:1702–1712, 2020.

[20] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[21] Q. Sun, J. Li, H. Peng, J. Wu, Y. Ning, P. S. Yu, and L. He. Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *Proceedings of the Web Conference 2021*, pages 2081–2091, 2021.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[23] Y. Xie, S. Li, C. Yang, R. C.-W. Wong, and J. Han. When do gnns work: Understanding and improving neighborhood aggregation. In *IJCAI'20: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI} 2020*, volume 2020, 2020.

[24] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[25] Q. Zhu, C. Yang, Y. Xu, H. Wang, C. Zhang, and J. Han. Transfer learning of graph neural networks with ego-graph information maximization. *Advances in Neural Information Processing Systems*, 34:1766–1779, 2021.

Deep Learning for Leakage Detection in Water Networks: A Comparative Study

Van Riemsdijk, Chris, and Julian Pasveer

Abstract—Water leaks in Water Distribution Networks (WDNs) are of paramount importance since water is a crucial resource for all life on earth. As we are growing as a population as a whole, it is necessary to have efficient management of water resources. This comparative study explores the state of the art regarding methods and algorithms used in order to detect water leaks in WDNs. All methods discussed in this comparative study use deep learning, where we focus on approach and methodology. We analyze, discuss and if needed discuss the methods for leakage detection. These methods consist of acoustic, pressure, and water flow sensor data. This data is fed to deep learning algorithms. The algorithms analyzed are deep neural networks, convolutional neural networks, and recurrent neural networks. We come to the conclusion that many of the aforementioned models have the capabilities to solve the leakage detection task, but they are highly dependent on the form that the input data takes.

Index Terms—Water Distribution Networks, Deep Learning, Deep Neural Networks, Leak Detection

1 Introduction

Leakage detection in water networks is paramount since water is a crucial resource for all life on earth. As we are growing as a population as a whole, it is necessary to have efficient management of water resources. Distributing water is done by water distribution networks (WDNs). These networks, consisting of water pipes, are mostly located underground and are therefore difficult to maintain. Due to, for example, the aging of the pipes or corrosion, leaks can form in these pipes. This can cause huge losses of water, resulting in economic problems. The percentage of the loss of water is between 15% to 50% [9]. Preventing water leakage is therefore a state-of-the-art research topic.

Deep learning has shown in multiple areas – computer vision, audio, natural language processing – that it can be a great estimator for many tasks. Where in classic machine learning (ML) we use a feature extractor, in deep learning (DL) feature extraction is done by the model itself. Due to this, a DL model is able to extract the important features from a WDN itself. Moreover, DL models contain the capability to model non-linear relationships for input vectors, which is the case for WDNs. Currently, ML and DL approaches exist for leakage detection. In the case of ML we observe artificial neural networks (ANNs) and for DL we notice recurrent neural networks (RNNs) such as long short-term memory (LSTM) models.

This paper will function as an analysis of the current state-of-the-art leakage detection models. These involve both machine and deep learning. The methods analyzed will be compared in order to find the best-fitting method/algorithm.

We begin by describing the state of the art regarding algorithms and methods that detect water leakages in WDNs in section 2. After that will follow a comparison between said algorithms and methods, which can be found in section 3. Finally, we will give our conclusion in section 4.

2 State of the art

In this section, we will go over the current state of the art regarding the algorithms and methods that have been invented in order to prevent water leakage in WDNs. In subsection 2.1 we will elaborate on leak detection systems that incorporate acoustics, whereas in subsubsection 2.2.1 the focus lies on pressure and water flow.

- *Chris van Riemsdijk is with Rijksuniversiteit Groningen, E-mail: c.m.van.riemsdijk@student.rug.nl.*
- *Julian Pasveer is with Rijksuniversiteit Groningen, E-mail: j.g.t.pasveer@student.rug.nl.*

2.1 Leak detection systems using acoustics

When a leak occurs in a pipeline, there are sounds that are emitted. These sounds are called acoustic emission (AE) signals. These signals, or waves, can be recorded by AE sensors. The signals that are being picked up can indicate leaks. In the past, researchers have used handcrafted features and created models from these AE waveforms to do leak detection [1]. Examples of such models are support vector machines (SVMs), and artificial neural networks (ANNs), next to models also tests are used such as Kolmogorov–Smirnov (KS) [8, 2]. These approaches are successful but do struggle with noise that is present in the data. However, as we observe in many areas of DL, humans are not capable enough to extract the proper features for a particular problem. This section will analyze two papers that came up with solutions that include the use of DL for detecting leakage in WDNs. The first paper, [10], introduces an AI-based method and compares three machine learning algorithms. The authors of the second paper, [1], decided to use DL to extract features from the data and to classify if leakage is present in the pipeline.

The Metropolitan Waterworks Authority (MWA), together with the National Electronics and Computer Technology Center (NECTEC), developed an AI-based leak detection device as well as a model that can filter out sounds that are created by water leakage from normal water flow in water pipe sounds. A platform was created, which consists of 4 parts. These include a smartphone, an acoustic rod/microphone, a cloud server, and a local server. The acoustic rod/microphone detects water leakage sounds, and sends the information to the smartphone, with whom a user can interact. The information is then sent to the cloud server.

2.1.1 Comparing machine learning algorithms

The leak detection method used is AI-based and can be separated into two phases. These are the training phase and the detection phase. The training phase needs input from the repair team, who provide the locations where sounds were collected. Both the target data and sound data are given to the machine learning algorithms in order to find the optimal model that accurately distinguishes leakage sounds from non-leakage sounds [10]. The detection phase sends the information, regarding whether it is actually a leak or not, to the maintenance team. The paper compares three machine learning algorithms. These include *Support vector machines*, *Deep neural networks (DNN)*, and *Convolutional neural network (CNN)*. Since SVM performed poorly, and over-

all is known to be less effective than DNN and CNN, we will not discuss it.
A DNN is an Artificial Neural Network (ANN), but has multiple hidden layers [10], resulting in a "deep" architecture, as can be seen in Figure 1. These hidden layers contain nodes, and all the nodes that

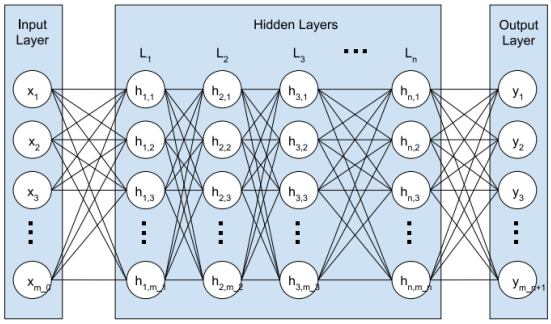


Fig. 1. DNN structure, image adopted from [10]

are in a specific layer are all connected to the following nodes in their next layer. This structure can therefore be called "fully connected layers" [10].
A CNN is different from an ANN in the sense that it has additional layers, which are called "convolutional layers". These layers can generate representative features of the given input data [10].

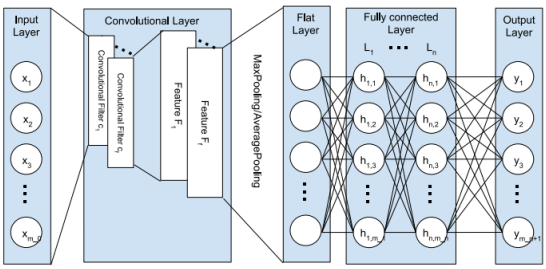


Fig. 2. CNN structure, image adopted from [10]

As can be seen in Figure 2, all the convolutional layers are generated by the input data. This is done using filters to generate features. CNN then uses the fully connected network for further learning features.
The data classification performance among DNN and CNN is done using real-world leakage and non-leakage sound data, which ended up being a total of 108,481 samples.

Table 1. Performance of DNN and CNN, data adopted from [10]

Training:				
Training	Method	Sensitivity	Specificity	Accuracy (%)
	DNN	0.9966	0.9974	0.9970
	CNN	0.9976	0.9961	0.9969
Testing				
	DNN	0.9374	0.9618	0.9489
	CNN	0.9317	99643	0.9471

2.1.2 Results

As can be seen in Table 1, the accuracy of both methods is very high and adjacent. Vanijjirattikhan et al. also generated confusion matrices in order to further compare the methods. A confusion matrix helps to visualize the true and false positives, as well as the true and false negatives. As can be seen in Table 2, DNN gave a good outcome with only a few true negatives and false positives.

Table 2. Confusion matrix DNN

Testing Data (DNN)		
	Positive	Negative
Positive	16175	1081
Negative	590	14856

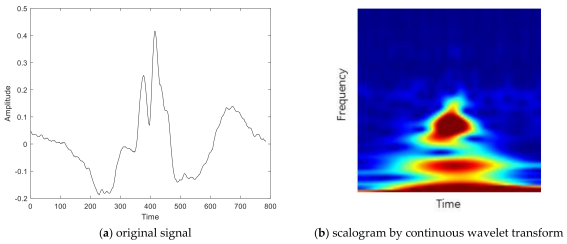
Table 3. Confusion matrix CNN

Testing Data (CNN)		
	Positive	Negative
Positive	16085	1179
Negative	551	14887

Table 3 shows that CNN, just like DNN, gives very good results with also few true negatives and false positives. As stated in Vanijjirattikhan et al., CNN and DNN give very similar results, but there can be a distinction between the two methods. Even though both methods give similarly good results, we have to take their complexity in mind. CNN is more complex than DNN and thus Vanijjirattikhan et al. decided to use the DNN algorithm for their maintenance activities and concluded that DNN was similar in accuracy in comparison with professional operators using conventional methods [10].

2.1.3 Leak Detection using Acoustic Emission waveforms

Ahmad et al. propose a pipeline to classify leak detection by AE waveforms. Before the data can be used from the AE sensors, the waveforms are transformed into images. This is done by a process called continuous wavelet transform (CWT). CWT is a process that takes the waveforms and transforms them into a time-frequency domain. [4]. These time-frequency scales result in a scalogram that can transform into a 3D image as seen in Figure 3



(a) original signal (b) scalogram by continuous wavelet transform

Fig. 3. Original signal to scalogram by CWT, image adopted from [3]

The proposed method was to use the final CWT image as input to the novel introduced pipeline containing a CNN and a convolutional autoencoder (CAE) resulting in a CNN-CAE pipeline. The CNN is built to extract local features, whereas the CAE will be able to construct the global features. These features are then merged and used as input to an ANN where classification is the final task to classify "Leak" or "No leak" turning this problem into a binary classification problem. The proposed pipeline can be seen in Figure 4

2.1.4 Global features: CAE

An autoencoder consists of 2 parts. An encoder and decoder. The function of an encoder is to extract the input to a latent space. This latent space is a low-dimensional representation of the input, then the decoder is used to upscale this low-dimensional representation to the original input image. Through this process, the mapping from input

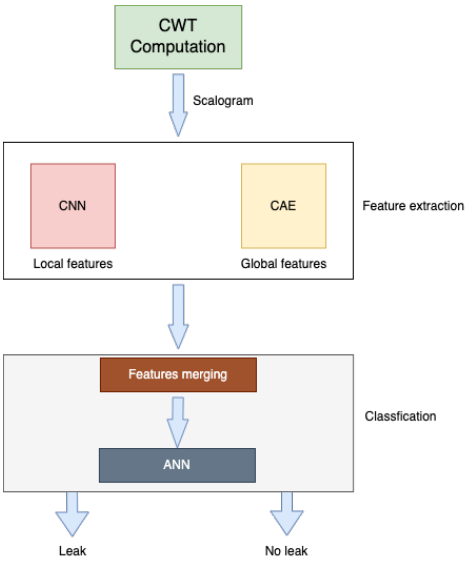


Fig. 4. Pipeline proposed by Ahmad et al.

Table 4. Performance of proposed and other methods on leak size of 0.3 mm and pressure of 7 bars. Data adopted from [1]

Leak size 0.3 mm, pressure 7 bar:

Metric	Proposed	FFT-CNN	CWT-LSTM	CWT-SVM (%)
Accuracy	**98.4%**	96.67%	90.1%	90.33%
Precision	**96.8%**	96.69%	91.0%	91.0%
Recall	**97%**	96.6%	90.5%	91.93%
F1 Score	**97.63%**	96.67%	90.7%	91.53%

Table 5. Performance of proposed and other methods on leak size of 0.5 mm and pressure of 13 bars. Data adopted from [1]

Leak size 0.5 mm, pressure 13 bar:

Metric	Proposed	FFT-CNN	CWT-LSTM	CWT-SVM
Accuracy	**96.67%**	93.33%	87.67%	95.33%
Precision	**95%**	94.0%	85.0%	94.1%
Recall	**95.27%**	93.33%	88.27%	93.33%
F1 Score	**95.3%**	93.27%	86.3%	94.27%

to latent is learned by the encoder, this is used by the authors of [1] to get the global features from the scalograms of the AE signals. The decoder is only used for training the CAE, as this will give the loss to back-propagate through the network and therefore give the correct feedback to the encoder part. The CAE proposed by Ahmad et al. has 4 convolutions combined with max-pooling layers, for the encoder. Then a flatten layer of size $(512, 1)$ which is the dimension of the latent space. The decoder is then the exact opposite, with 4 transposed convolution layers. returning to the original image dimensions. At the end of training this CAE, we end up with a network that produces a feature vector of size 512.

2.1.5 Local features: CNN

The CNN proposed by Ahmad et al. is really similar to the CAE. It has the same four convolutional layers with a combined max pooling layer. The shape in the final is, $(8, 8, 8)$ this means that the final layer has a 512 feature vector. As said previously, the CNN in this model is used to extract the local features from the scalogram. The rectified linear unit (ReLU) is used as an activation function throughout the layers to introduce non-linearity. The authors of [1], that are using this architecture, will extract the local features.

2.1.6 The final step: classifying

From the CAE and CNN, we get a combined feature vector of size 1024. This is then forwarded to a shallow artificial neural network (ANN). This shallow network has an input layer with 1024 nodes, 1 hidden layer of 512 nodes, and an output layer of 2 nodes representing leak or no-leak. Both input and hidden layers introduce dropout to add regularization to the model. Finally, we use the "SoftMax" activation function and the binary cross-categorical entropy loss to train the network.

2.1.7 Results and discussion

For the experiment, Ahmad et al. used its own experimental industry setup where they would artificially create data samples that have leaks. The data were obtained at pressures of 7 and 13 bar. 240 samples for each pressure, respectively. Where 50% contained leaks in the time series and 50% were no leak samples. Then n-fold cross-validation was performed on 70%, 30% splits for $n = 10$. The proposed method was compared with 3 other methods: FFT-CNN, CWT-LSTM, and CWT-LSTM. Where FFT is the fast Fourier transform. The results are seen in Table 4 and Table 5

As we can observe, the proposed method by Ahmad et al. outperforms all other reference methods. The authors claim that this is due to the more extensive and better feature extraction compared to the other methods. Although this could be true, there is some skepticism from us due to multiple reasons. First, the dataset only contained 480 samples, for a DL model this is little. Especially when the data is created in an artificial environment where the data can contain similarities between every sample. This leads to a non-diverse dataset which, of course, deep learning models can easily adapt to if the data is too similar. Moreover, the implementation details of the other reference methods are not given and do not allow for reproducibility and might raise questions if the architectures are tuned to the designed problem. However, the results seem promising for the proposed problem.

2.2 Leak detection using data from WDN features

Sound is not the only feature to detect leakages, a WDN has many features about the water flowing through its network. Literature suggests that by looking at features from the WDNs we can also extract useful information to detect and sometimes localize leaks. Two of the most prominent features to observe are water pressure and water flow. The distinction between the two is subtle but important. Pressure relates to the force that flows through a pipe, whereas flow relates to the amount of water in that same pipe.

In this section, we will review and discuss two different papers in the respective areas. Javadiha et al. propose a pipeline that finally can detect leaks via a CNN pipeline. Whereas Lee and Yoo use historic time series data in combination with a recurrent neural network (RNN) to create thresholds for the real data to detect leaks.

2.2.1 Leak localization using pressure measurements

Certain leakage detection methods make use of pressure sensors. This is also used in Javadiha et al. Water pressure sensors are cheaper than conventional flow sensors and, moreover, are easier to install. Furthermore, there are also fewer pressure sensors necessary than conventional ones. Javadiha et al. propose a method that uses Deep Learning for analysis and exploration through the map representation of pressure residuals of the WDN. Javadiha et al. assume that the leak localization step is done correctly by an efficient method and hence does not care much about this part.

2.2.1.1 Proposed method

The proposed method can be found in Figure 5.

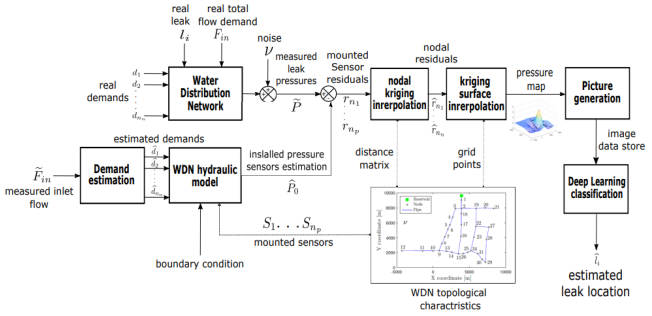


Fig. 5. Leak localization scheme proposed by Javadiha et al., image adopted from [6]

The scheme itself serves as an overview of the proposed method. For the sake of the scope of this paper, we are only interested in the final step, which is the Deep Learning Classification step. Javadiha et al. propose to use a Convolutional Neural Network for image classification. To make sure that precise classification is achieved, Javadiha et al. created a framework architecture for CNN.

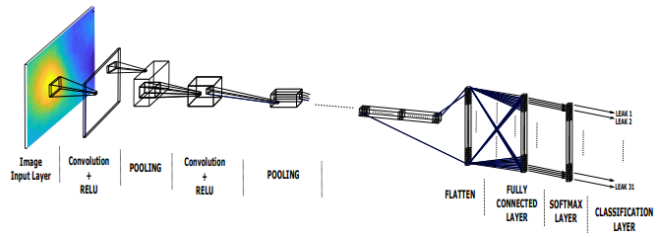


Fig. 6. CNN architecture, image adopted from [6]

This architecture is often seen in CNNs since the architecture of a CNN usually consists of (several of) these layers. Javadiha et al. describe that the data training set has to be split into batches. The CNN will then train through these data batches, after which it shuffles the data. This is done in order to avoid over-fitting. Furthermore, to enhance the accuracy, Javadiha et al. make use of the Bayesian Theorem and apply it recursively.

The training process was done using the Deep Learning Toolbox of Matlab. A data set consisting of 20 days of samples was used. The CNN operation was validated using 15% of random data [6]. When the CNN model was calibrated, the testing could start. Javadiha et al. make use of a confusion matrix (Γ) in order to assess the results. The rows are used to indicate the leak scenario, whereas the columns show which leak is located by the leak localization method [6]. Such a matrix its accuracy can be assessed by the following equation:

$$Ac = \frac{\sum_{i=1}^{n_n} \Gamma_{i,i}}{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{i,j}}$$

Furthermore, [6] make use of the Average Topological Distance (ATD). This is used in order to assess the leak localization performance in WDNs. The ATD is the average value of the minimum distance which is in between the nodes that have a leak and the node candidate proposed by the leak localization method. The ATD can be calculated with the following equation:

$$ATD = \frac{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{i,j} A_{i,j}}{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{i,j}}$$

[6] put all the obtained values of the Accuracy and the ADT in a table to check the results. The results can be seen in Table 6. As is clear from the table, as soon as the number of sensors increases (as well as the time H), the accuracy increases and the ATD decreases. As is clear from these numbers, the results are promising. [6] do address the fact that their method its accuracy might be vulnerable to large amounts of nodes and uncertainties, as could is often the case in real WDNs.

Table 6. Obtained Accuracy and ADT performance indicator results. Data adapted from [6]

	Accuracy		ADT	
Num. of sensors	H=1	H=24	H=1	H=24
4	47.27	56.14	1.487	0.9969
5	56.09	66.26	0.8625	0.5269
6	74.33	81.41	0.3297	0.1994
7	75.59	81.05	0.2823	0.1465
8	77.91	89.2	0.2751	0.1080
10	82.21	91.58	0.2168	0.0842
12	87.81	94.13	0.1438	0.0586

2.2.2 Leakage detection based on inflow meter data

Lee and Yoo state that leaks in water networks can affect consumers for at least 12 hours to several weeks in Korea. Primary tasks such as draining, and restoring, these networks are easy and require little to no time. However, the essential task of accurately detecting leak accidents while also being able to pinpoint the exact leak position is still an ongoing process. A reliable leak detection system should have the properties mentioned earlier, while also raising little to no false alarms. Therefore, the authors decided to go for a data-based deep learning approach based on recurrent neural networks (RNNs) combined with a multi-threshold model to have more robustness to false positives. The specific model, called "long short-term memory" (LSTM) is from the RNN family. LSTM is popular as it solves the vanishing gradient problem for RNNs and is able to use previous input to make predictions.

2.2.2.1 Approach

Lee and Yoo focus their research on water flow. Water flow is the amount of water that is going through the WDN pipes. A measure for this is cubic meters per day (CMD). The authors propose a four-step pipeline; 1) Data preprocessing. 2) Flow prediction using the deep learning method RNN-LSTM. 3) Creating boundaries from the predicted flow using multi-threshold classification. 4) From this we can do real-time leak recognition. We will go through them one by one in the following subsections. However, this pipeline enables the following graph to be constructed:

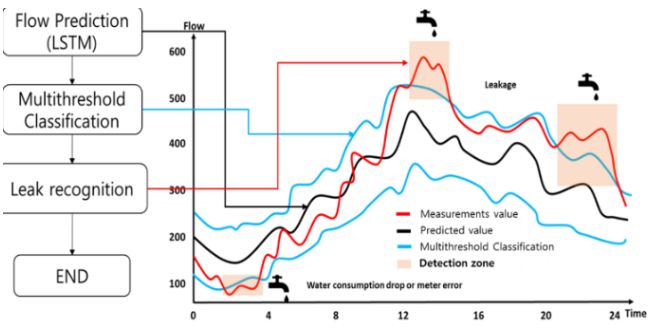


Fig. 7. Graph of all flows determined by the data from step 1. The multi-threshold classification is based on the predicted values from the RNN-LSTM. The measurement value is the value captured in real-time. The detection zones show where the measured value is higher or lower than the thresholds. Image adapted from [7]

2.2.2.2 Data preprocessing

As we are dealing with time-series data over multiple days, the authors used data preprocessing. The data consists of 5-minute readings per

day with their corresponding CMD. To train the model, the data is split up into 3-day readings, where every 5 minutes of the previous day will predict the 5 minutes for the day that is to be predicted. An example of this can be seen in Table 7.

Table 7. Performance of proposed and other methods on leak size of 0.3 mm and pressure of 7 bars. Data adapted from [1]

Example of data as input to the model

Time	Day 1	Day 2	Day 3		Prediction day
0:05	3	4	1	→	?
0:10	4	4	3	→	?
0:15	2	3	3	→	?
0:20	5	4	4	→	?

2.2.2.3 Flow prediction using RNN-LSTM

As said earlier, Lee and Yoo use a RNN-LSTM model to predict the water flow in the WDN. The LSTM cell has risen in popularity in the last few years, as it solves the gradient vanishing problem that RNNs mostly have. Moreover, due to its architecture, it is able to give meaning to historical data, known as context. This is helpful as we use this historical data in the training and inference steps to predict the water flow. An example of a LSTM cell can be seen in Figure 8

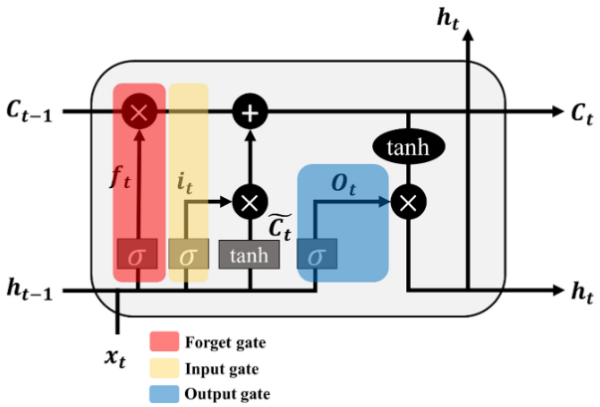


Fig. 8. LSTM cell architecture, image from [7]

The three highlighted regions show how the LSTM cell can keep track of historical data with its internal weights. This is done in the forget-, input- and output-gate. These gates have internal weights that are trainable and thus can manage how much weight is given to historical, input, and output data.

2.2.2.4 Multi-threshold classification

From the preprocessed data, a LSTM model was trained to make predictions for 5-minute segments. This yields a vector of predictions over a single day. However, the real ground truth of the measured water flow lies around this prediction. Therefore, we need to create a threshold region where the measured value is still "acceptable", or in other words, there is no leak as seen in Figure 7. There are multiple ways to do this in the literature but Lee and Yoo chose to use the $\bar{X}$ chart method with different confidence intervals (99%, 95%, 90%). The paper does not go into detail about which interval was the best, however, they state that empirically the best confidence interval was chosen.

2.2.2.5 Results

To get the best model hyperparameters the authors did an empirical search for the best parameters. For this, a metric is needed. The metric that was used to compare certain hyperparameters was: mean average precision error (MAPE). MAPE is given by the following formula:

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{X_t - \hat{X}_t}{X_t}\right|$$

A lower MAPE score results in a better model. Lee and Yoo found that 120 epochs and a batch size of 60 showed the best results. The authors trained the model on real data from a WDN in South Korea. Their training set consisted of the days between the 13th and 22nd of January. Then the test data was only from the 23rd of January. The authors split the data up into 6 zones. The results show that 5 of the 6 zones had an accuracy of leak detection over 90%, and the remaining zone had an accuracy of around 40%. The authors explain that the cause of the low accuracy in one of the zones is the shortage of data. As the authors use real data and only had 10 total days of data, this is understandable, but this might implicate that the other zones are performing well while overfitting has happened.

3 Discussion & Comparison

Multiple deep learning architectures have been used to detect and sometimes localize water leakages. The papers discussed in this comparative study either used pressure, flow, or acoustics as input data to feed to the deep learning models. The deep learning models that have been used are CNN, CAE, DNN, and RNN-LSTM. In this section, we want to discuss and/or compare the model architectures from the literature study.

3.1 Convolutional neural networks

Convolutional neural networks have shown in many areas that they are quite proficient in solving tasks. Examples of such areas are vision and audio. Many of these tasks involve an image as input. Observing the papers about acoustic emission and water pressure, we see that CNNs are mostly used to extract the local feature maps from the images as input. As [1] creates scalograms by applying a continuous wavelet transform on the sound data and then combining that with the global features of the CAE. This enables the authors to create a DNN with the combined features from the CNN and CAE as input. Whereas [6] creates a 2D representation of the pressure map created from the preprocessing steps and uses the low-level features from the CNN in a DNN as well to classify leaks. The CNN architecture proposed by [1] had promising results, and their method outperformed all other methods they compared it with. However, the data set used was artificially generated and this could have impacted the results, mainly because the data set used contained only 480 samples. [6] propose a method consisting of a pipeline that also used a CNN. The results were promising since high accuracy levels were obtained. The problem with their approach however is the fact that the data set that was used only contained 20 days' worth of samples. [6] state that for larger data sets, their method might end up getting unsatisfactory results, as the accuracy could be vulnerable to a higher number of nodes.

3.2 DNN

As seen in the previous section, DNNs are not only used as the single solution to a task but are also used in other architectures to do predictions from the low-level feature maps created by for example a CNN. In combination with being computationally less expensive compared to other deep learning architectures, it forms a powerful tool. As was shown in subsection 2.1, [10] was the only study in this literature review that actually used a DNN as the main architecture. It stood up to the CNN introduced in the same paper. Therefore, being the best model selected in that study as compared to CNNs, it is computationally cheaper. The other studies showed that using a DNN is great for using low-level feature maps to add extra non-linearity to improve prediction performance.

3.3 RNN-LSTM

RNN-LSTM is a great tool when it comes to time series forecasting. As mentioned in paragraph 2.2.2.3, RNN-LSTM is able to have trainable – thus adjustable – parameters when it comes to giving weight to historic input. Lee and Yoo put this to the test with their setup. Although they achieved accuracies above 90% they suffered from a low amount of real data; only possessing 10 days of 5-minute readings. The problem with having such little data is that the results can be misleading as the model is not tested against other dates in the year, as perhaps a higher demand in a warm summer could increase water flow in the WDN.

3.4 Discussion

Combining the findings of all models and approaches, we observe that there is no golden rule, model, or architecture to detect and/or localize leakages in water distribution networks. The state-of-the-art in section 2 shows that there are multiple answers to the problem. For both sound-related and water-related data, the authors in the different studies find and explore different ways of pre-processing data, thus resulting in different input vectors for the deep learning architectures. This makes comparing the methods with each other hard or not even possible, as comparing model performances on different tasks is ineffective. However, there are some common possible problems that we see in most of the reviewed literature. Because leakage detection using deep learning is new, there is little to no public data available. Comparing this to ImageNet [5], a database with millions of images, many studies tend to artificially create leaks, or even go to the extent to model the data itself. We think that the shortage of quality data results in a plethora of different solutions, and that when there is more real data available, research will be more focused on a LSTM or transformers/attention-based [11] approach.

4 Conclusion

Water leakages in WDNs can result in significant water losses, reduced water pressures, and in certain cases, it can lead to damage to infrastructure. In this study, we have analyzed three different deep learning architectures, namely CNNs, DNNs and RNN-LSTMs which all seem to have an application in leakage detection. Comparing the methods, we observe that the before-mentioned architectures have the capability to solve the problem of detecting water leakages. This is mostly due to the nature of the problem that is presented by the studies done in the literature. Different datasets lend to different solutions. Processing your data to the final input of your model has many implications on what model to use. Images are great for convolutional neural networks as they can achieve low-level feature maps that humans can not extract, however, RNN-LSTMs are better at the task of time-series forecasting. This means that the raw data and processing of this data leads to an architecture that suits the problem the best.

Acknowledgements

We would like to thank D. Düştegör for her assistance during this project.

References

[1] Sajjad Ahmad et al. “A Method for Pipeline Leak Detection Based on Acoustic Imaging and Deep Learning”. en. In: *Sensors* 22.4 (Feb. 2022), p. 1562. ISSN: 1424-8220. DOI: 10.3390/s22041562. URL: https://www.mdpi.com/1424-8220/22/4/1562 (visited on 02/13/2023).

[2] Nawal Kishor Banjara, Saptarshi Sasmal, and Srinivas Voggu. “Machine learning supported acoustic emission technique for leakage detection in pipelines”. In: *International Journal of Pressure Vessels and Piping* 188 (2020), p. 104243. ISSN: 0308-0161. DOI: https://doi.org/10.1016/j.ijpvp.2020.104243. URL: https://www.sciencedirect.com/science/article/pii/S0308016120302192.

[3] Yeong-Hyeon Byeon, Sung-Bum Pan, and Keun-Chang Kwak. “Intelligent Deep Models Based on Scalograms of Electrocardiogram Signals for Biometrics”. In: *Sensors* 19.4 (2019). ISSN: 1424-8220. DOI: 10.3390/s19040935. URL: https://www.mdpi.com/1424-8220/19/4/935.

[4] Yiwei Cheng et al. “Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural network”. In: *Knowledge-Based Systems* 216 (2021), p. 106796. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2021.106796. URL: https://www.sciencedirect.com/science/article/pii/S0950705121000599.

[5] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[6] Mohammadreza Javadiha et al. “Leak Localization in Water Distribution Networks using Deep Learning”. In: *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. 2019, pp. 1426–1431. DOI: 10.1109/CoDIT.2019.8820627.

[7] Chan-Wook Lee and Do-Guen Yoo. “Development of Leakage Detection Model and Its Application for Water Distribution Networks Using RNN-LSTM”. In: *Sustainability* 13.16 (2021). ISSN: 2071-1050. DOI: 10.3390/su13169262. URL: https://www.mdpi.com/2071-1050/13/16/9262.

[8] Akhand Rai and Jong-Myon Kim. “A novel pipeline leak detection approach independent of prior failure information”. In: *Measurement* 167 (2021), p. 108284. ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2020.108284. URL: https://www.sciencedirect.com/science/article/pii/S0263224120308241.

[9] Harshit Shukla and Kalyan Piratla. “Leakage detection in water pipelines using supervised classification of acceleration signals”. In: *Automation in Construction* 117 (2020), p. 103256. ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2020.103256. URL: https://www.sciencedirect.com/science/article/pii/S0926580519310301.

[10] Rangsarit Vanijjirattikhan et al. “AI-based acoustic leak detection in water distribution systems”. In: *Results in Engineering* 15 (2022), p. 100557. ISSN: 2590-1230. DOI: https://doi.org/10.1016/j.rineng.2022.100557. URL: https://www.sciencedirect.com/science/article/pii/S2590123022002274.

[11] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Machine Learning for Leak Detection in Water Networks

Xiaoyu Guan (s3542807) Lonneke Pulles (s3533603)

Abstract— In many countries, water is distributed by so-called Water Distribution Networks (WDNs) via pipes to homes and businesses, but not all water that enters these systems reaches its designated goal. Water that trickles out of the network before it reaches the consumer is called non-revenue water. Sometimes more than 25% of a country's water production can be non-revenue, rich countries not excluded.
This paper contains an overview of seven state-of-the-art methods to detect water leaks in WDNs with machine learning methods applied to different kinds of sensor data, categorized into three different approaches. We will discuss the advantages and disadvantages of each method, and judge them based on their costs, the degree of inclusion of physical theory in each model, efficient sensor placement, versatility regarding different network topologies, performance in various leak scenarios, and computational complexity.
It is found that the decision tree-based machine learning methods such as Random Forest and CatBoost perform best across almost all sensor types when taking both performance and computational complexity into account. However, more research is needed to more accurately compare the exact costs, time complexities and performances of different machine learning techniques on the same real-life datasets.

Index Terms—Smart city, Environmental technology, Hydroinformatics, Leak detection, Machine learning, Water distribution networks

1 Introduction

Recent years have seen a rise in applications of machine learning to a multitude of domains. In combination with the decline in sensor costs, this has paved the way for smart cities.

One vital responsibility of cities is the distribution of water to its inhabitants and businesses. Water is often transported via a city's underground pipeline system called a water distribution network (WDN), but these systems are not foolproof. When water that enters the system leaks out of the system before it can reach its designated goal, it is called non-revenue water. This can be a significant portion of water production, as in some countries more than 25% of water is non-revenue, not excluding richer countries.

The problem is that leakages are often hidden and difficult to detect, causing the leaks to exist for a long time. They endanger community health due to the ingress of impurities in the water and result in substantial water and economic losses. In the city of Hong Kong, the fraction of non-revenue water was at 16% in 2016, amounting to a financial cost of US$173 million per year [6][17]. Moreover, undetected leakages may cause underground cavities and even sinkholes. Hence, effective leak detection methods are needed to increase the sensitivity of leak detection and detection speed [8].

Nowadays, automated leak detection is an active research field. Several methods have been proposed based on different measurement techniques, such as acoustic emission, fiber optic sensors, pressure point analysis, negative pressure waves, and ground penetration radars [8]. Moreover, many statistical, machine learning and deep learning have been proposed for leak detection over the last few decades, steadily increasing the number of papers published in the field.

A reliable classification or predictive model is needed to build a successful detection system. Machine learning, which is a subfield of artificial intelligence, has proved its effectiveness in classification and regression to predict outcomes [3]. Classification problems are often tackled by supervised machine learning techniques [16], of which the most common include Artificial Neural Networks (ANN), Bayesian Networks (BN), Decision Trees (DT), k-Nearest Neighbors (KNN), Logistic Regression (LR), Random Forests (RF) and Support Vector Machine (SVM). In recent years, deep learning methods such as deep neural networks (DNN) and Long Short-term Memory (LSTM) have been added to the field.

Though some review papers have attempted to clarify the differences between many leak detection methods, to our knowledge no practical and applicable recommendations are given.

This paper provides an overview of state-of-the-art methods to detect leaks in water distribution networks. Seven papers that propose different kinds of machine learning-based detection methods were selected. They were explored in detail and compared against each other, based on six aspects such as robustness in various leak scenarios, the balance between costs and performance, and applicability to a variety of water network topologies. After comparing these six characteristics, we provide a flowchart with simple decision criteria for water network managers that aim to decide which leak detection method is most suitable for their specific business case. With the flowchart, we aim to answer the question: what is the most suitable machine learning-based leak detection system for a given water distribution network?

This paper is structured as follows. Section 2 will cover background information on the four machine learning methods and the sensors that provide the data, after which we will describe the methods of the four papers in more detail in section 3. Next, section 4 comprises the results, and section 5 includes the discussion. A final summary is provided in section 6 and suggestions for future directions of research are posed in section 7.

2 Background

Before more sophisticated leak detection methods came into existence, leak detection could be done with the so-called WEC rules, invented by Western Electric Company in 1958 [12]. They are a set of four statistical rules. Operators of the water management company had to measure the water pressure over various time intervals. When any of the WEC rules were violated, an alarm was sounded.

In the previous decades and during the last few years, leak detection has increased in complexity and sophistication. Depending on the different detection methods that are used to detect a leak, a leak detection system can be categorized as a passive or an active system. Passive systems detect leaks via human vision and manual sensor utilization, and are the most traditional way to identify leaks. Active systems report leaks by analyzing the data collected by sensors [5], such as noise loggers, flow sensors, and accelerometers. These systems can be divided into three distinct approaches, namely transient-based, model-based, and data-driven [5]. A hierarchical overview of current water leak detection technologies is shown in figure 1. In this paper, we only focus on active systems.

- *Xiaoyu Guan is a MSc Computing Science student at the University of Groningen, E-mail: x.guan.1@student.rug.nl.*
- *Lonneke Pulles is a MSc Computing Science student at the University of Groningen, E-mail: l.c.pulles@student.rug.nl.*

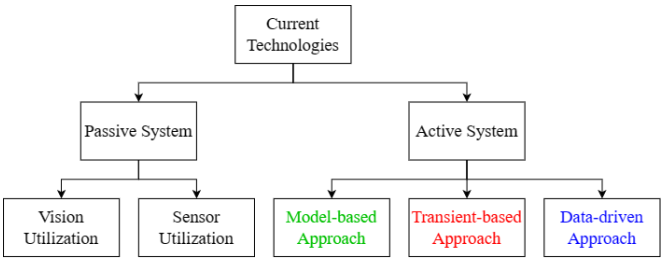


Fig. 1: Hierarchical categorization of state-of-the-art water leak detection technologies, adapted from [5].

2.1 Model-based approach

The model-based approach is a popular approach used in modern water leak detection systems based on a hydraulic model [11]. This model is a comprehensive simulation of the network created with detailed knowledge of its physical quantities, such as a pipe's roughness and hydraulic radius. Once constructed, it needs to be calibrated to ensure its prediction match reality. Leak detection is then done by comparing the predictions to the measurements.

2.2 Transient-based approach

Another possible approach to leak detection is based on pressure transients. These are pressure waves that are sent through the pipeline. A schematic representation of a pipeline with a transient source is shown in figure 2.

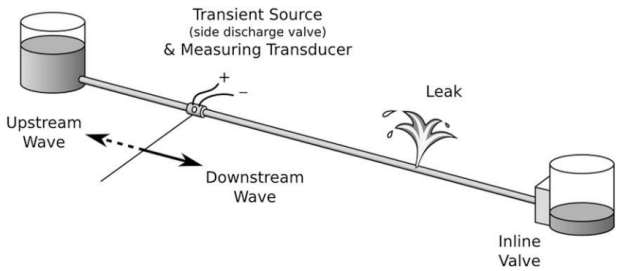


Fig. 2: Schematic representation of a pipeline with the generation of leak-reflected signals in a transient trace [6].

Whenever the wave encounters an obstruction, junction, or other change in the network, the wave pattern is altered. These alterations are modeled upon the inception of the leak detection system to create a base state. A leak causes the wave pattern to deviate from the expected pattern [6]. A visualization of such a deviation in pressure, also called a head response, is shown in figure 3.

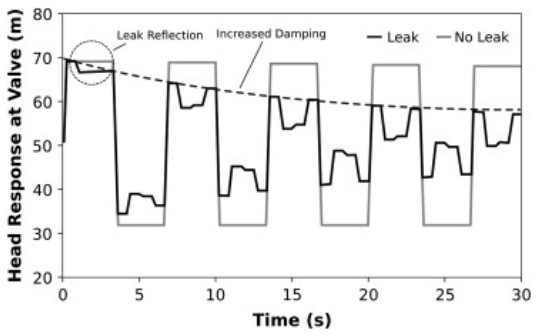


Fig. 3: Transients in leak and no-leak pipelines [6].

2.3 Data-driven approach

The data-driven approach is another approach that is increasingly used nowadays. The idea of this approach is to detect the leak by directly analyzing the data collected from the sensors [11], without simulating the network. Because the data-driven approach heavily relies on historical data needed for training machine learning models, the quality of the collected data is important [5].

2.4 Network topology and sensor placement

The topology of a water distribution network concerns the junctions and edges of all water pipelines in the network, modelled as graphs. An example topology of the water distribution network of the Italian city Modena is shown in figure 4. This large-scale network is formed by 268 junctions connected through 317 pipes and served by 4 reservoirs [13].

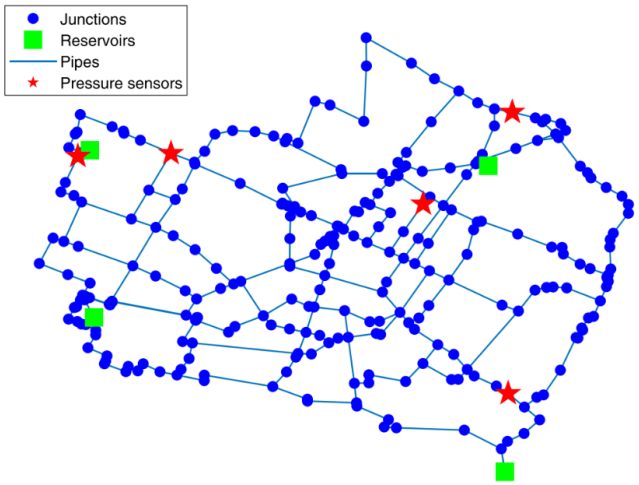


Fig. 4: Schematic representation of the water distribution network in the Italian city Modena [13].

The topology of a water network is especially important during sensor placement. Optimal sensor placement for leak detection in WDNs has been studied extensively in the field of hydroinformatics. Optimal sensory locations, however, are distinctly different for different types of sensors [12]. In general, it holds that more sensors also lead to higher accuracies, a higher sensitivity to smaller leaks, and a decrease in the time until detection.

The differences between the optimal placement of pressure and flow sensors is due to hydraulics. For pressure sensors, the most informative location is at the end of the network, because that is the location where pressure will drop most in case of a leak. On the other hand, flow sensors should be placed in pipes near the source of the network [12].

Hagos et al. [12] found that when around 10% of the pipes in a water network are monitored with flow meters, a true positive rate of almost 100% was reached. The study used a relatively simple statistical detection method invented in 1958 based on the so-called WEC rules. On the other hand, pressure meters at most reached a true positive rate of 82% according to the same authors. The false positive rate of flow meters, however, is higher than that of pressure sensors. For pressure meters it is relatively constant between 10% and 20%, whereas for an increasing number of flow meters the false positive rate can increase up to 96%. When installing flow meters, it is therefore good to keep in mind that a lower number of sensors could actually result in a better performance.

3 Method

3.1 Paper selection

To obtain a selective overview of the most important methods in the research field, we selected papers that proposed model-based, transient-

based and data-driven approaches for leak detection. For each category, we chose two papers and we attempted to select some of the most recently published papers in the field. Additionally, we also selected one paper using a hybrid approach with deep learning, due to the high popularity of deep learning methods nowadays.

Next to the papers proposing specific methodologies, we also used two review papers on model-based versus data-driven methods [11] and on transient-based methods [2] to obtain a more comprehensive overview of the field, despite the limited selection of papers.

An overview of the reviewed papers can be found in table 1.

Table 1: An overview of the reviewed papers.

Reference	Approach	Feature	(Best) ML method
Fereidooni (2021) [9]	Model-based	Flow	Bayesian network / Random forest
Rejeesh (2019) [15]	Model-based	Pressure	Random forest
Levinas (2021) [7]	Transient-based	Pressure transients	KNN
Asghari (2023) [1]	Transient-based	Pressure transients	CatBoost
Fares (2022) [8]	Data-driven	Vibrations	Random forest (train) / NN, DL and SVM (validation)
Tariq (2021) [17]	Data-driven	Vibrations	Random forest
Quiñones-Grueiro (2021) [13]	Model-based & Data-driven (hybrid)	Flow & pressure	Deep NN

3.2 Model-based approaches

We focus on two model-based approaches, proposed by Fereidooni et al. [9] and Rejeesh et al. [15].

3.2.1 A hybrid model-based method for leak detection in large scale water distribution networks [9]

Fereidooni et al. [9] proposed a model-based method that is applicable to relatively large water distribution networks. They tested four different machine learning algorithms, KNN, Random Forest, Decision Trees and Bayesian networks, on both real and simulated data. They found that Bayesian networks were the most effective and robust, reaching high accuracies and F-scores. KNN performed worse on accuracy and F-score, but could handle the imbalanced training data better. Random forest was more robust, but its performance degraded when the data was imbalanced. Decision trees had good precision, but bad recall.

3.2.2 Random Bagging Classifier and Shuffled Frog Leaping Based Optimal Sensor Placement for Leakage Detection in WDS [15]

Rejeesh et al. [15] proposed a method called Random Decision Tree Bagging Classifier based Shuffled Frog Leaping Optimization (RDTBC-SFLO) and compared it with a one-dimensional convolutional neural network and support vector machine (1D-CNN-SVM) and multi-objective ant colony based optimization (ACO) model. They found that 1D-CNN-SVM and ACO have significant shortcomings in practice. RDTBC-SFLO consists of two parts: classification and optimization.

- **RDTBC.** The decision trees were constructed with the iterative Dichotomiser 3 (ID3) algorithm that applied a top-down greedy approach to achieve the binary classification task of detecting a leak or no leak.
- **SFLO.** SFLO is an optimization algorithm that was used to find the best position to place the sensor in order to minimize the number of sensors used in the water network.

The data used to train the model was based on pressure measurements and the new method RDTBC-SFLO has a big improvement that increases by 24% compared to the 1D-CNN-SVM model and 9% compared to the multi-objective ACO model [15].

3.3 Transient-based approaches

We focus on two transient-based approaches, proposed by Levinas et al. [7] and Asghari et al. [1].

3.3.1 Water Leak Localization Using High-Resolution Pressure Sensors [7]

The dataset was generated by simulating water leakages, each of which contains the ID for each pipe, the high-pressure value series, the distance from the pipe's start, and the diameter of each leak [7].

In order to generate the dataset that the KNN model uses, the authors first specified the leak location and diameter, then applied the hydraulic transient simulation to define the leak properties such as leak discharge, and set the transient simulation as pressure-driven demand.

Levinas et al. [7] showed that the method they used can not have a good performance when the network becomes too complex, but for simpler networks with less than 10 nodes the results are acceptable.

3.3.2 Machine learning modeling for spectral transient-based leak detection [1]

The method proposed by Asfhari et al. [1] used a transient-based leak detection methodology using the CatBoost machine learning model. CatBoost is an ensemble method of oblivious decision trees. The model was trained on more than 3.8 million synthesized data records for classifying leaky sections and predicting sizes. The authors used the machine learning model as a substitute for more traditional optimization methods.

The authors also considered various other machine learning methods. Random forests were considered to take too much computation time on large datasets, and another decision tree-based model called CART was said to easily overfit on the training data. Similarly, SVM and KNN were said to take too much computation time on considerably large datasets. Because the dataset was imbalanced regarding leak and no-leak scenarios, the authors also thought neural networks and logistic regression were inappropriate due to the way their cost function is implemented and their approach to minimizing the average error rate. CatBoost according to the authors performs better on imbalanced datasets in less time.

The model was compared to XGBoost, Linear and Logistic regression and Genetic Algorithms, and outperformed all of them on leak localization accuracy, recall and computational time needed. Only linear and logistic regression needed a shorter training time.

3.4 Data-driven approaches

We focus on two data-driven approaches, proposed by Fares et al. [8] and Tariq et al. [17].

3.4.1 Data-driven application of MEMS-based accelerometers for leak detection in water distribution networks [17]

Tariq et al. used a dataset that was collected in the Hong Kong water supply network by placing Micro-Electro-Magnetic-Sensors (MEMS)-based accelerometers on the gate valves or fire hydrants near potential leak locations from October 1st, 2020 to July 31st, 2021. All the data was validated by the Hong Kong water supply department [17]. MEMS accelerometers combine the advantages of MEMS technology, which have higher accuracy than other sensors because of the 3D microstructure, and the benefits of accelerometer technology. Accelerometers can detect water leakage without damaging the pipes, since an accelerometer can simply be placed on the surface [4, 17]. Figure 5 shows an example of how an accelerometer is placed on the

Fig. 5: Place an acceleration on gate valve[17]

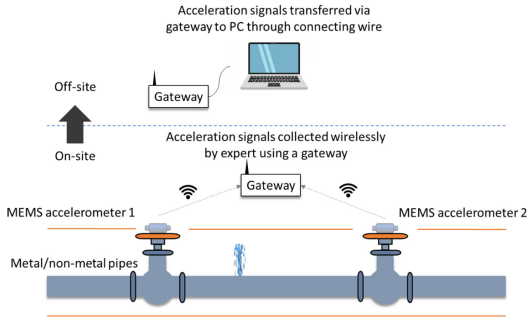


Fig. 6: Signal data transformation[17]

surface of a gate valve. Figure 6 shows the communication between the sensor and a PC.

After the data was collected successfully, the authors first applied a signal processing algorithm to the collected data to preprocess it and get rid of noise.

The machine learning methods they used to detect the water leak were k-Nearest Neighbors (KNN), Decision Tree (DT), (RF) and AdaBoost.

As is shown in the results below, Random forest performs best in both metal-based and non-metal-based models. In addition, the two ensemble-based machine learning models (RF and AdaBoost) have no significant difference in the performance of metal-based and non-metal-based models. However, the two individual-based machine learning models (KNN and DT) have dramatically worse performance in non-metal-based models. The results can be found in table 2.

Table 2: The training dataset results on the metal-based model and non-metal-based model proposed by Tariq et al. measured in percentages [17].

	KNN	DT	RF	AdaBoost
Metal	96.72	99.18	100	99.18
Non-metal	89.86	84.78	94.93	94.2

4 Results

In this section, we present the advantages and drawbacks of the selected methods regarding six factors: the degree of preprocessing and modelling needed, applicability to various network topologies, performance in various leak scenarios, additional possibility in leak localization and size estimation, sensor placement and the resulting costs and spatial resolution we can achieve in localization, and computational complexity.

4.1 Degree of modelling

The degree of hydraulical modelling in a leak detection model can have a profound effect on an approach's accuracy. Measurement and approximation errors propagate when modelling is not handled well. In order to prevent this from happening, the models need to be carefully calibrated to ensure predictions of flow and pressure represent the reality. A disadvantage is therefore that whenever the topology of a network changes, for example due to the addition of an extra pipe to the network, the entire model needs to be reconstructed. This means that if a water supply company implements a model-based system, it needs to hire experts that can adjust the model accordingly whenever the need arises.

Another disadvantage in the use of physical quantities is that they can change over time. For example, the roughness of a pipeline can decrease due to erosion. When these kind of changes are not observed and the model is not recalibrated, the prediction can increasingly diverge from reality [11].

4.2 Performance in various leak scenarios

A big challenge in data-driven methods based on acoustic sensor data is the variability in the sound emitted by different kinds of leaks. The pipe vibrations depend amongst others on the pipe material, leak shape and size, water pressure and pipe diameter. Moreover, background noises coming from vehicles driving on the roads above the underground pipelines can distort signals. To overcome these difficulties, the sensor data needs to be preprocessed with various time- and frequency-based methods, such as Fourier and wavelet transform [8]. Additionally, the data-driven method proposed by Fares [8] generally performed better on metal pipelines than on nonmetal (e.g. plastic) pipelines.

4.3 Sensor placement

The proposed method of Fereidooni et al. [9] installs flow sensors in each junction of the network and does not consider optimal sensor placement. On the other hand, Rejeesh [15] and Quiñones-Grueiro [13] use different optimization methods to find optimal sensor placements.

The transient-based methods by Levinas [7] and Asghari [1] make no mention of sensor placements. The acoustic and data-driven methods by Fares [8] and Tariq [17] mention placing accelerometers on easy to reach locations like valves, and do not take optimal sensor placement into account either.

4.4 Leak localization

An advantage of sound-based and vibration-based sensor systems is the possibility to additionally localize leaks based on the time lag of the vibrations between the two sensors that are installed at each end of a pipeline [10]. Localization was nevertheless not implemented into the acoustic methods proposed by Fares et al. [8] and Tariq et al. [17].

In contrast, the method proposed by Quiñones-Grueiro et al. [13] based on both flow and pressure sensors also performs localization and estimation. Nevertheless, the localization method only returns a pair of nodes that are at the ends of the pipeline with the supposed leak. The localization method is necessary because the method uses only 9 sensors in the network depicted in figure 4 containing 268 nodes. The prerequisite for these next two steps is that the leak detection is very accurate.

The model-based method proposed by Fereidooni et al. [9] is able to localize the leak, as sensors are placed on each junction of the network. The model-based method proposed by Rejeesh et al. [15] is also able to localize the leak, but with less sensors.

In transient-based methods [7][1], leak localization is an inherent part of leak detection.

4.5 Costs

Concrete costs are never mentioned in any of the papers that were reviewed, but some indications of prices are given. Quiñones-Grueiro et al. [13] indeed mention that an alternative to the popular flow sensor-based systems is based on pressure sensors, as they are less expensive and easier to install and maintain, but more are needed to reach the same performance [12]. In contrast, according to Fereidooni et al. [9], their method based on flow meters is actually the cheaper sensor system to install.

Another possible sensor system measures vibrations in the pipelines with accelerometers or noise loggers, which are wireless sensor systems that communicate remotely and that are based on accelerometers.

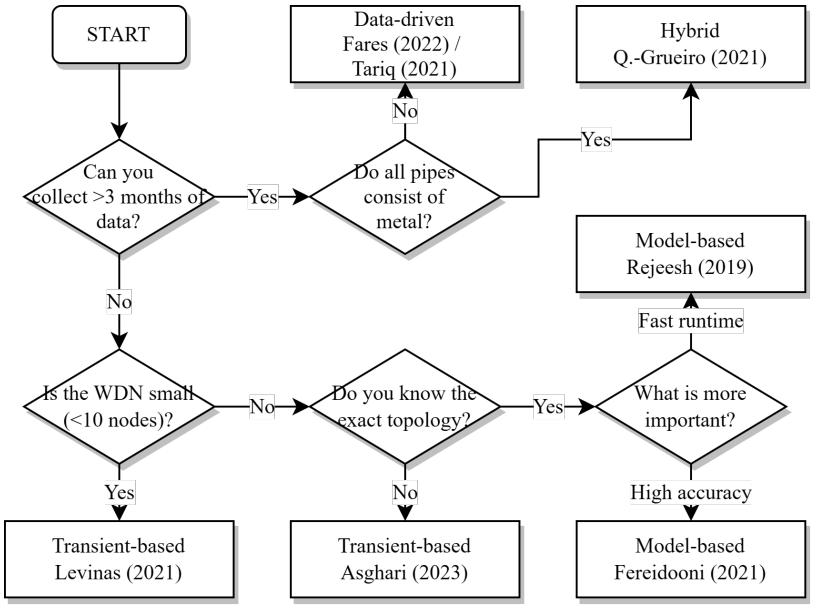


Fig. 7: A flowchart to support decision making when choosing a sensor system for leak detection in a specific water distribution network.

The acoustic method proposed by Fares [8] is according to its authors 'inexpensive', but again no concrete prices were given. Some authors [9][13] on the other hand mention that noise logger-based systems, such as proposed by Van Hieu et al. [10], are expensive to deploy.

Next to sensor prices, the cost of a leak detection system can also be influenced by the detection algorithm. Tariq et al. mention that both training and deploying a CNN-based detection system is 'expensive' [17]. When sensor data is for example obtained every 15 minutes [13], this means that the detection algorithm also needs to run every 15 minutes. The power supply costs for both the algorithm and the sensors are therefore important to take into account, but electricity prices vary per country and are not mentioned in any of the papers. What we can research to obtain a comparison of power consumption by proxy, is the algorithm's time complexity.

4.6 Computational complexity

Unfortunately, the same problem as when one tries to study the costs of each proposed method, occurs when one tries to review the computational complexities, albeit in a lesser fashion. Most papers do not give exact time complexities or execution times, and often only qualifiers like 'impractical computation time' or 'computationally expensive' are given as an indication.

One exception is the paper by Asghari et al. [1], in which the authors mention that their method based on CatBoost, which they say is faster than popular methods such as Random Forest and neural networks, takes 15 minutes to train and its computation time for prediction is less than 0.01 seconds. In comparison, the hybrid deep learning-based method [13] needs in the order of seconds to estimate leak size, on a much smaller network.

4.7 Which method to use in practice

In order to give specific recommendations for an engineer wishing to implement a leak detection system in their water distribution network, we look at the factors taken into account above and other review studies [11][2]. We use the fact that according to these review studies, a data-driven approach is more appropriate when a large amount of historical data can be obtained from a real network. However, when there is less data and a network's hydraulic model is easy to obtain, model-based methods are preferred [11].

Figure 7 contains a flowchart that proposes a decision making process a water supply department or company can go through when deciding on a leak detection system.

5 Discussion

A major downside when comparing studies in the field of leak detection in WDNs with machine learning is the lack of public datasets. Because obtaining real-life data is only available to the management companies of the specific water distribution networks, researchers that do not have access to real-life measurements have to simulate them with (expensive) software or try to reproduce real-life settings in a lab. This means that even when a study reports a relatively good accuracy on its machine learning method, there is a high likelihood that this accuracy would not uphold in the real world.

As a result, and because each study used vastly different datasets of different WDNs which were both measured in the real world and simulated with software, it is impossible to compare methods from different papers based on quantitative performance metrics. In addition, because there is no publicly available dataset containing data from all four mentioned sensor types, we could not properly compare and reproduce the studies on our own computers. We can only compare the machine learning methods as they were performed within each study.

6 Conclusion

In conclusion, machine learning methods can have a positive impact on detecting water leakage. As we reviewed the seven papers in the previous sections, we found that decision tree-based machine learning methods such as Random Forest and CatBoost seem to have the best performance for the most common types of pipes and many kinds of sensor data, while also taking low computational complexity into account. However, more research is needed to more accurately compare the exact costs, time complexities and performances of different machine learning techniques on the same real-life datasets.

7 Future work

The research into ML-based detection methods for leak detection can be extended into methods that also perform estimation of the leak size, such as has been done with deep learning methods by [13], and that offer a more precise localization of the leak.

Because decision tree-based methods often outperformed other machine learning techniques, future research can survey if different decision tree-based ensemble methods have better performance on the leak detection problem than existing methods. For example, an improvement on random forest is called probabilistic random forest [14]. This method could be tested and compared to other existing methods in the future.

One current problem in the research into automatic leak detection methods is the lack of publicly available datasets based on real data. As a result, many researchers result to simulation software to create their own data, which are often based on different network topologies, sensor types and leak scenarios. Only a dataset that contains data from a vast range of network configurations could provide a thorough comparison study of different leak detection methods based on quantitative measures. We suggest that the creation of such a benchmark problem could speed up the research in the field by simplifying the data collection process. The water company Vitens started such an initiative in 2015, but a more comprehensive dataset would be beneficial in the future. Moreover, in 2018 some researchers aimed to solve this exact problem by publishing a comprehensive benchmark problem called LeakDB [18], but this dataset was obtained from simulations and has not yet been widely adopted by researchers. Nevertheless, it is a step in the right direction.

Acknowledgements

The authors wish to thank Elnur Seyidov and Koen Bolhuis for their useful feedback, and are especially grateful for the helpful advice and mentoring by Dr. Dilek Düştegör.

References

[1] V. Asghari, M. H. Kazemi, H.-F. Duan, S.-C. Hsu, and A. Keramat. Machine learning modeling for spectral transient-based leak detection. https://www.sciencedirect.com/science/misc/pii/S0926580522005568, 2023.

[2] A. Ayati, A. Haghighi, and H. Ghafouri. Machine learning approach to transient-based leak detection of pressurized pipelines: Classification vs regression. https://doi.org/10.1007/s13349-022-00568-2, 2022.

[3] J. E. Black, J. K. Kueper, and T. S. Williamson. An introduction to machine learning for classification and prediction. https://pubmed.ncbi.nlm.nih.gov/36181463/, 10 2022.

[4] R. Bogue. Mems sensors: past, present and future, 2007.

[5] T. K. Chan, C. S. Chin, and X. Zhong. Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection. https://ieeexplore.ieee.org/iel7/6287639/6514899/08565861.pdf, 2018.

[6] A. F. Colombo, P. Lee, and B. W. Karney. A selective literature review of transient-based leak detection methods. https://www.sciencedirect.com/science/misc/pii/S1570644309000094, 2009.

[7] G. P. Daniel Levinas and A. Ostfeld. Water leak localization using high-resolution pressure sensors. https://www.mdpi.com/2073-4441/13/5/591, 02 2021.

[8] A. Fares, I. A. Tijani, Z. Rui, and T. Zayed. Leak detection in real water distribution networks based on acoustic emission and machine learning. https://www.tandfonline.com/doi/abs/10.1080/09593330.2022.2074320, 2022. PMID: 35506881.

[9] Z. Fereidooni, H. Tahayori, and A. Bahadori-Jahromi. A hybrid model-based method for leak detection in large scale water distribution networks. https://link.springer.com/article/10.1007/s12652-020-02233-2, 02 2021.

[10] B. V. Hieu, S. Choi, and Y. Kim. Wireless transmission of acoustic emission signals for real-time monitoring of leakage in underground pipes. https://link.springer.com/misc/10.1007/s12205-011-0899-0, 2011.

[11] Z. Hu, B. Chen, W. Chen, D. Tan, and D. Shen. Review of model-based and data-driven approaches for leak detection and location in water distribution systems. https://iwaponline.com/ws/misc-pdf/21/7/3282/1104597/ws021073282.pdf, 04 2021.

[12] M. H. D. J. K. E. Lansey. Optimal meter placement for pipe burst detection in water distribution systems. https://iwaponline.com/jh/misc/18/4/741/30092/Optimal-meter-placement-for-pipe-burst-detection, 2016.

[13] M. Quiñones-Grueiro, M. A. Milián, M. S. Rivero, A. J. S. Neto, and O. Llanes-Santiago. Robust leak localization in water distribution networks using computational intelligence. https://www.sciencedirect.com/science/misc/abs/pii/S0925231221001442, 2021.

[14] I. Reis, D. Baron, and S. Shahaf. Probabilistic random forest: A machine learning algorithm for noisy data sets. https://iopscience.iop.org/article/10.3847/1538-3881/aaf101/pdf, dec 2018.

[15] S. G. Rejeesh Rayaroth. Random bagging classifier and shuffled frog leaping based optimal sensor placement for leakage detection in wds. https://link.springer.com/article/10.1007/s11269-019-02296-7, 2019.

[16] A. Singh, N. Thakur, and A. Sharma. A review of supervised machine learning algorithms. https://ieeexplore.ieee.org/abstract/document/7724478, 2016.

[17] S. Tariq, B. Bakhtawar, and T. Zayed. Data-driven application of mems-based accelerometers for leak detection in water distribution networks. https://www.sciencedirect.com/science/misc/pii/S004896972106188X, 2022.

[18] S. G. Vrachimis, M. S. Kyriakou, et al. Leakdb: a benchmark dataset for leakage diagnosis in water distribution networks. https://ojs.library.queensu.ca/index.php/wdsa-ccw/article/view/12315, 2018.

Implementation of Active Queue Management Algorithms on Programmable Network Switches: A Review

Stern Brouwer and Florian de Jager

Abstract— Bufferbloat is a common problem that occurs when there is excessive buffering of packets in a switch. This buffering can cause significant delays in the transmission of packets, ultimately leading to reduced overall network performance. To address this issue, several Active Queue Management (AQM) algorithms have been proposed, which aim to regulate queue sizes and prevent bufferbloat. These algorithms work by monitoring queue length and selectively dropping packets to avoid congestion. However, finding the optimal buffer size is crucial, as excessively small buffers may result in packet loss and reduced throughput. This paper provides a comprehensive overview of Active Queue Management (AQM) algorithms for programmable switching ASICs. The study discusses the following AQM algorithms: Controlled Delay (CoDel), Random Early Detection (RED), Proportional Integral Enhanced (PIE), Fine-Grained AQM (FG-AQM), (Dual) Proportional Integral 2 (PI2), ingress RED (iRED) and Proportional Integral Controller Enhanced (PIE). The implementation details of each algorithm are analyzed, and the appropriate use cases for each algorithm are determined. The research also highlights the advantages, challenges, and performance of the AQM algorithms. We find that AQM algorithms are crucial for preventing bufferbloat and improving network performance, and programmable switching ASICs, such as Intel Tofino, offer high-speed processing that can be leveraged to run AQM algorithms. Among the algorithms studied, iRED and FG-AQM are reported to be highly performant. However, some of the algorithms deviate from RFC standards due to architectural constraints in the ecosystem. This study will help researchers and practitioners better understand and select the most appropriate AQM algorithms for their specific applications.

Index Terms—Active Queue Management, TCP, Tofino, Bufferbloat

1 Introduction

The rapid growth of network traffic calls for the need for efficient network operation. Over the years, the volume of data transmitted over networks has grown exponentially, driven by the proliferation of mobile devices, streaming video services, and cloud-based applications. Cisco reported a growth from an average of 6 exabytes (6 million terabyte) per month in 2009 to 42 exabytes in 2014 [1]. In 2022, the amount of data sent is around 396 exabytes per month. To support the transmission of this massive volume of data, packet processors play a key role. They are responsible for handling the thousands of packets that traverse the network every second, routing them to their intended destinations and performing any necessary processing or filtering along the way. However, as the volume of network traffic continues to grow, so too does the potential for congestion and delays.

To combat this issue, the Transmission Control Protocol (TCP) can be leveraged, as it is already a widely used protocol for the transmission of data over the Internet. One of the key features of TCP is its congestion control mechanism, which is designed to prevent network congestion by avoiding over utilization and ensuring fairness among competing flows [2]. TCP's congestion control mechanism employs a variety of techniques, including slow start, congestion avoidance, and fast retransmit, to regulate the rate at which data is transmitted over the network and prevent network congestion.

However, excessive buffering of packets at the egress port of a forwarding switch or router can cause a related problem called bufferbloat, which delays the transmission of packets and reduces overall network performance. To address this issue, various AQM algorithms have been proposed to regulate queue sizes and prevent bufferbloat. AQM algorithms work by monitoring the length of queues and selectively dropping packets. On the other hand, if the buffers get too small, packets may be dropped, leading to lost data and reduced throughput. So AQM algorithms must try to find the balance in the buffer size. There exist a variety of different AQM algorithms, such as Controlled Delay (CoDel) and random early detection (RED). These methods, among others, are discussed in detail in Section 2.

Programmable switching ASICs (Application-specific integrated circuits), such as the Intel Tofino [3], are good candidates for running AQM algorithms as they are programmable and offer speed beyond those of traditional programmable switches. Tofino has a maximum processing capacity of 12.8 Tb/s [3], which is significantly higher than traditional Field Programmable Gate Arrays (FPGAs) that can only achieve a throughput of up to 900 Gb/s [4].

The main contribution of this study is to provide a comparative overview of state-of-the-art AQM algorithms for programmable switching ASICs. Our research will be organized around three main contributions: (1) Providing a comprehensive overview of AQM algorithms for programmable switching ASICs, (2) Identifying and analyzing the implementation details of each algorithm, and (3) Determining the appropriate use cases for each algorithm. By achieving these objectives, this study will contribute to the current research landscape in AQM for programmable switching ASICs, helping researchers and practitioners better understand and select the most appropriate AQM algorithms for their specific applications.

The paper is structured as follows: Section 2 provides background information relevant to the research topic, followed by a discussion of the methodology used to conduct the research in Section 3. Section 4 presents the main approaches found in the literature review. The research questions are answered in Sections 5 and 6, which discuss the advantages, challenges, and performance of some AQM algorithms. Finally, the conclusion and future work are presented in Section 7.

2 Background

This section provides an overview of AQM and the most relevant algorithms. Moreover, it explains the details about programmable switching ASIC along with its architecture model and programming languages that can be used.

2.1 Active Queue Management

AQM algorithms operate at the switch level by actively managing the length of network queues to prevent packet loss and control latency. If the queue gets too long, the latency will be too high and bufferbloat occurs, if the queue is kept too short, packets may be dropped, leading to lost data and reduced throughput. AQM algorithms are needed to

- *Stern Brouwer is with the University of Groningen, E-mail: s.c.brouwer.1@student.rug.nl.*
- *Florian de Jager is with the University of Groningen, E-mail: f.q.de.jager@student.rug.nl.*

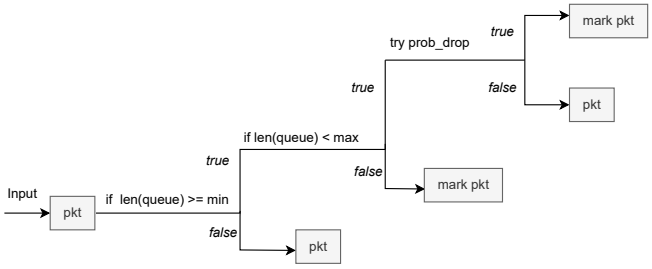


Fig. 1: Simplified RED flowchart that determines whether a packet is marked or not. [5]

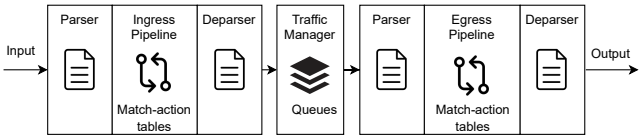


Fig. 2: Simplified PSA architecture that shows how the packets are processed [15].

manage this balance between latency and throughput. Congestion occurs when the network traffic exceeds of the capacity of the network, resulting in dropped packets and increased latency. AQM algorithms are designed to mitigate the effects of congestion by, e.g. dropping or marking packets, so that the sender reduces their rate of transmission. Several AQM algorithms have been developed over the years, each with different approaches. These include RED [5], CoDel [6], PIE (Proportional Integral Controller Enhanced) [7], PI2 [8], and DualPI2 [9]. In the following subsections, these algorithms will be discussed in a bit more detail.

2.1.1 Random Early Detection

RED is one of the first AQM algorithms. It operates by marking packets with a probability of being dropped, based on the queue length [5]. The marking probability is calculated using an exponentially-weighted moving average of the queue length over time. It has two thresholds: a minimum and a maximum. A simplified flowchart is shown in Figure 1. In the figure, if the end result is "mark pkt" the packet is marked for drop, if the result is "pkt" they are not marked. If the queue length is below the minimum threshold, packets are not marked. If the queue length is above the maximum threshold, packets are marked for drop. If the queue length is between the minimum and maximum thresholds, packets are randomly marked for drop with a probability that increases as the queue length approaches the maximum threshold.

2.1.2 Controlled Delay

Although RED seemingly solved the bufferbloat problem at first, it became clear that RED's configuration parameters were difficult to set, i.e. its parameters are interdependent and sensitive to the traffic characteristics of the network. Moreover, RED did not perform well in several network scenarios, leading to reluctance to use it [10]. Moreover, research showed that queue length was not a reliable predictor of congestion, which RED relied on to trigger packet drops [11]. This lead to the development of CoDel, and it was proposed as an improvement over RED [6]. CoDel operates by measuring the queuing delay instead of queue length, and it drops packets in increments to maintain a low and stable queue delay. Furthermore, CoDel introduces a differentiation between "good" and "bad" queues and this idea is based on the function of queues in the network. In the network, queues can act as buffers to absorb bursts of traffic [6]. CoDel's approach is to not react to the queues that are able to recover themselves within one round trip time (RTT). On the other hand, when the delay exceeds a certain number of RTTs, it is considered a sign of a "bad" queue, indicating that packets are experiencing excessive delays. CoDel then intervenes to control the queuing delay in a timely and efficient manner.

2.1.3 Proportional Integral Controller Enhanced

PIE randomly drops incoming packets when congestion is detected [7]. However, unlike the RED algorithm which detects congestion based on the queuing length, PIE detects congestion based on the queuing latency, using the derivative (rate of change) of the queuing latency to help determine congestion levels and appropriate response. PIE is designed to maintain the benefits of RED, including easy implementation and scalability to high speeds while addressing the latency issue. PIE is designed to ensure high link utilization without suffering link underutilization or losing network efficiency, and it is almost as simple to implement as RED [5].

2.2 The Programmable switching ASIC

The programmable switching ASIC is a fairly recent innovation in the research space of Software-defined Networking (SDN). Traditional switches have their scheduling algorithms fixed from the point that they are manufactured, whereas programmable switching ASICs can be reprogrammed to use a different scheduler. One example of a programmable network switch is the Intel Tofino [3]. These novel switches give the server operator finer-grained control over the network traffic, and this innovation has also led to a lot more research, specifically implementing an AQM on the programmable switch [12]. Next, the basic architecture of programmable switches will be explained.

2.2.1 Programmable switch architecture

There exists a couple of different models in the literature: Banzai [13], PISA (used for P4-14) [14] and PSA (used for P4-16) [15] all propose a different, but similar enough model. Figure 2 highlights a basic representation that shows the functions that are present in the PSA model. This figure shows the flow of packets through a pipeline. Initially, incoming packets undergo parsing, where relevant header fields are extracted. The deconstructed header is then processed by match+action tables in the ingress pipeline, allowing for actions that modify the header contents. In these tables, certain values in the header (or based on internal variables stored in the switch), are matched, and are accompanied by a corresponding action. Before the packets are buffered into a queue, the packets are deparsed, so the (altered) headers are put back to create a transmissible packet. The packets in the queue are now awaiting dequeuing by the traffic manager. Once dequeued, the packets pass through the egress pipeline, where additional actions can be applied before transmission, like in the Ingress pipeline [14, 15]. The amount of actions that are available in the Match+action tables in the egress differs from the actions available in the ingress, this influences the design of different AQM algorithms.

2.2.2 Programming languages

One key feature of programmable switches is the ability to program them using high-level languages. P4 is a language that has been specifically designed for configuring switches that are programmable, as referenced in [16]. P4 is protocol-independent, meaning it is not tied to specific packet formats. The language operates by modifying packet header specifications, allowing for the customized processing of packets. Currently, there are two relevant versions of P4: P4-14 and P4-16, both of which have their own specifications [14, 15]. P4 is not the only language that can be used for programmable switches, Domino, is an example of another one [13]. Domino has more C-like syntax, so is higher level than P4.

3 Research Methodology

Our main research question is: How do Active Queue Management Algorithms compare when implemented on programmable network switches? We also define the following sub-questions:

RQ1 What are the advantages and challenges of implementing AQM algorithms on programmable network switches?

Paper	Score	Type	RQ
Tofino + P4: A Strong Compound for AQM on High-Speed Networks? [17]	B	TP, EX	RQ1
P4-CoDel: Active Queue Management in Programmable Data Planes [18]	7	TP	RQ1
Active Queue Management on the Tofino programmable switch: The (Dual)PI2 case [12]	A2	TP, EX	RQ1
iRED: Improving the DASH QoS by dropping packets in programmable data planes [19]	B4	TP, EX	RQ1 RQ2
Fine-Grained Active Queue Management in the Data Plane with P4 [20]	8	TP, EX	RQ1 RQ2

Table 1: Evaluation of papers.

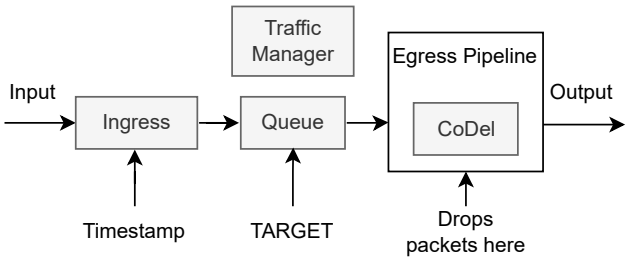


Fig. 3: The P4 reference pipeline with CoDel [18].

RQ2 What are the performance gains of using AQM algorithms on programmable network switches?

3.1 Finding literature

To find literature that is relevant to our research questions, we performed a rapid review. We chose this method as a lighter alternative to the systematic review due to time constraints. We used the RUG library search engine (SmartCat) to search and gather articles. We constructed our search query in an iterative way, where we added keywords from our topics that helped us get more relevant results. We eventually narrowed down the following query: "(AQM OR "Active Queue Management") AND ("programmable network switch" OR "Programmable Switch" OR "Tofino" OR P4)", which gave us 87 results.

3.2 Evaluation of articles

In Table 1, all articles that have been found and used in this paper are listed along with the journal score, research type, and the relevant research question. The research type has been abbreviated with the following labels: experiment (EX), technical paper (TP), literature review (LR), and survey (S).

4 What do the papers propose

In this section, we will give an overview of the various AQM algorithms that are proposed in the papers. These AQM algorithms already had software based implementations, but had previously never been implemented on a programmable data plane.

4.1 CoDel

We start with CoDel, which was one of the first rewrite of an AQM algorithm to work on the P4 language, published in 2018 by Kundel et al. [18]. They were able to integrate it into the P4 reference pipeline. It is situated in the egress part of the pipeline and targets the buffer unit just before it, thus being able to drop packets, shown in Figure 3.

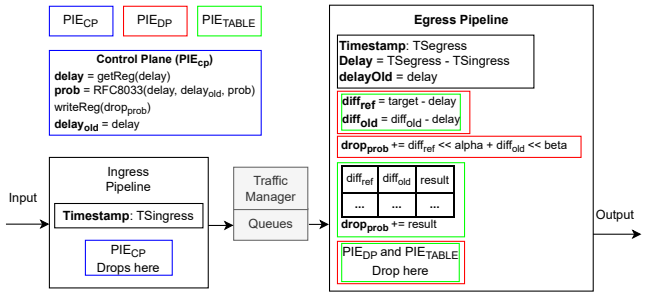


Fig. 4: The P4 reference pipeline with PIE_{CP}, PIE_{DP}, and PIE_{TABLE}. [17].

The CoDel algorithm that has been implemented here is similar to the original. The algorithm has two parameters, TARGET and INTERVAL, and ensures that the queuing delay periodically stays below the TARGET value. If the delay is below TARGET, no packet is dropped. If the delay exceeds TARGET by more than INTERVAL, the first packet is dropped, and the interval between dropping packets decreases until the TARGET delay is reached. There have been some necessary modifications to make it work in P4, which we will go over in section 5.

4.2 PIE

Next, the PIE algorithm is discussed, which has been implemented on P4 to run on the Tofino by Kunze et al [17]. They proposed three flavors that each have their own advantages and drawbacks.

The first is the implementation PIE_{CP} which is illustrated in figure 4, where the data plane drops a packet based on the drop probability at the ingress and measures the queuing delay for forwarded packets at the egress. The control plane periodically samples the queuing delay, updates the drop probability, and writes it back to the ASIC using the basic functionality, scaling, and exponential decay according to the RFC [7].

PIE_{DP} is the second PIE algorithm port that provides approximate results for drop probability computations. Unlike PIE_{CP}, it does not include drop probability update scaling and exponential decay, which makes it more aggressive when congestion is low and slower to react when congestion decreases.

PIE_{TABLE}, the third of the PIE algorithms, replaces PIE_{DP}'s bit shift drop probability approximations with table lookups of precomputed probabilities, which increases its precision. The performance of PIE_{TABLE} depends on the precision of the table entries, which is related to the number of stored key-value pairs.

4.3 PI2 and DualPI2

Another set of algorithms that have been ported to the P4 language is DualPI2 and (single-queue) PI2 by Gombos et al. [12]. In addition to this, they also made an effort to run the P4 implementation on the Intel Tofino switch. In figure 5, we can see the PI2 algorithm in the egress part of the P4 reference pipeline.

The P4 code for PI2 calculates the queuing delay for each packet and stores it in a register. The control plane periodically uses the delay to calculate the probability factor for marking TCP traffic. The updated probability is stored in two registers and used to determine whether to mark or drop TCP traffic.

The DualPI2 algorithm on the other hand requires two queues and uses weighted round-robin scheduling. It prioritizes L4S traffic and can sacrifice some throughput during overload to ensure a minimum throughput for normal traffic. The congestion signals for the two queues are coupled, with a stronger signal in the L4S queue. The ingress part forwards timestamps and implements a classifier to assign traffic to the corresponding queue. The P4 reference pipeline of DualPI2 works similarly to PI2 in figure 5.

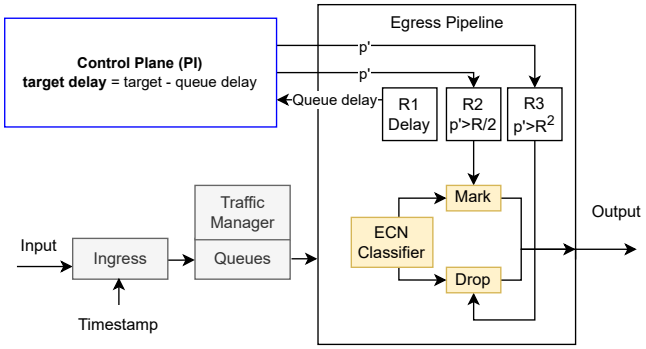


Fig. 5: The P4 reference pipeline with PI2 [12].

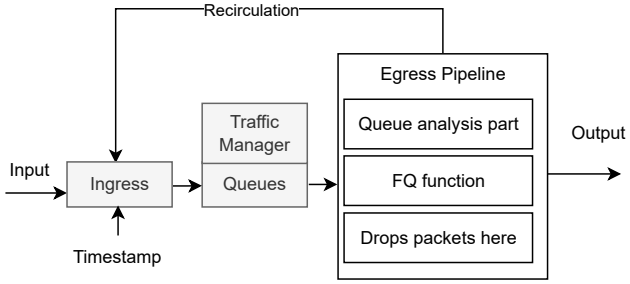


Fig. 6: The P4 reference pipeline with FG-AQM [20].

4.4 Fine-Grained AQM

Another algorithm that has been proposed is Fine-Grained AQM (FG-AQM) [20]. The FG-AQM system consists of two modules: the queue state analysis module and the drop probability calculation module. The first module analyzes the queue state, focusing on calculating the target flow and reducing the impact of tailed flows. The second module calculates packet loss probability based on the network state and uses a proportional integral control model to adjust the drop probability. This system effectively utilizes the trend of network state changes to improve network performance.

Figure 6 shows the FG-AQM algorithm implemented using the P4 reference pipeline. The algorithm is deployed mainly in the Egress and includes a queue state analysis module and a drop probability module. The match-action part defines which data flow will start the algorithm. The packet processing in PISA includes recursive and clone methods to update the tailed flows counter in the flow analysis part and reduce the performance impact of tailed flows.

4.5 iRED

Last, we have an algorithm called iRED, which has been implemented on P4 by Almeida et al. [19]. Specifically, it proposed a modification to the RED algorithm and called it iRED (ingress RED), capable of dropping packets at the ingress pipeline. Compared to some AQM strategies that were discussed before in this paper, it drops packets directly at egress.

In the egress pipeline, as shown in Figure 7, the iRED algorithm decides whether to drop a packet or not, based on the average queue size and dropping probability. If the packet is not dropped, it is sent to the next hop, while a cloned packet is recirculated to the ingress pipeline to indicate queue congestion. In the ingress pipeline, the iRED algorithm drops the next packet for a particular output port with the drop flag ON and turns the flag OFF, preventing future packets from exacerbating the queue buildup.

5 Advantages and Challenges

This section presents the findings of all papers that could give a piece of the answer to the research question: What are the advantages and challenges of implementing AQM algorithms on programmable network switches?

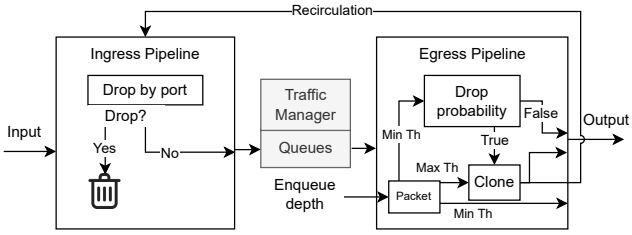


Fig. 7: The P4 reference pipeline with iRED [19].

5.1 The incentive of AQM

The papers we found all agree that there is a need for programmable ASICs that are able to run various kinds of AQM algorithms that have been proposed over the years. Traditional, non-programmable, ASIC switches often come with RED [5], which is hard to parameterize properly [21]. These ASIC switches are most often used at places such as data centers, where lots of network traffic needs to be routed and throughput is most important. The downside of ASIC switches is that they are hard to design and produce, which makes integrating new AQM schemes expensive. That is where programmable ASIC switches come in, which have been in development for some time. Languages, such as P4, allow porting of new or existing AQM algorithms to make use of this hardware directly.

5.2 Drawbacks of AQM on programmable switches

There do exist several drawbacks and limitations to the P4 ecosystem, including the Intel Tofino that can run it. The first are complex arithmetic functions, such as square root, which some algorithms need [18, 17]. A workaround for this can be an approximation for the function, where we store the values of $\sqrt{n}$ in the P4 table. It was determined by Kundel et al. [18] that this does not significantly harm the performance of the CoDel algorithm.

P4 does not have an iteration construct and loops can only be created by the parser state machine [15]. The language also does not support recursive functions, and as a result, the work performed by a P4 program depends linearly only on the header sizes.

There is also a constraint on how operations in P4 are mapped to a finite number of match-action units (MAUs) and how each packet can only pass through an MAU once, with the P4 compiler responsible for synthesizing the P4 code to a set of MAUs and finding a suitable layout that can be mapped onto the switch's resources. The allocation of resources may be a challenge even if the P4 code is valid and target-specific [17]. This placement problem becomes even harder when you take into account that memory such as maps and registers is stage-local. Thus, it can only be accessed and assigned to one MAU.

Most AQM algorithms that we found to have been ported to P4 also drop packets at the egress part of the P4 reference pipeline. This means that the buffer contains packets that are potentially going to be dropped but have to go through the entire switch [19]. The iRED algorithm is able to alleviate this problem by splitting the dropping and decision-making parts into ingress and egress, respectively.

There also exist some platform-specific drawbacks to the Intel Tofino. These limitations arise from the ASICs pipeline operation mode, which consists of a programmable packet parser, several stages of match-action units, a buffer, and a traffic manager, where AQM algorithms would typically reside. However, the traffic manager is not programmable using P4, which is a limitation to implementing AQM algorithms [12].

Another limitation, also on Tofino, is that of arithmetic operations such as multiplication. P4 itself only allows the multiplication of unsigned integers, but the Tofino is much more constrained than that. Numbers must be either defined from a table lookup or as a compile-time constant [17].

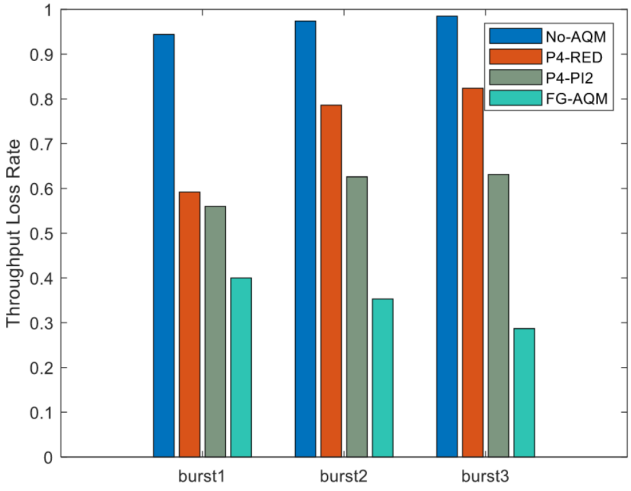


Fig. 8: Throughput Loss Rate as reported by [22].

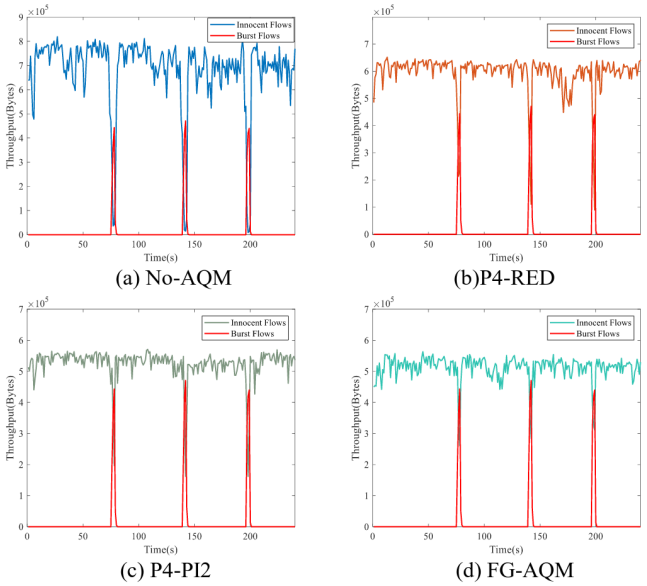


Fig. 9: Influence of burst flows as reported by [22].

All these constraints mean that it is not trivial to port AQM algorithms to target P4-enabled switches such as the Tofino. AQM designs often depend on specialized functionality, not on a most common subset of functionalities.

6 Performance of the Algorithms

This section presents performance metrics of the AQM algorithms on programmable network switches of all papers that have been found, which can answer the research question: What are the performance gains of using AQM algorithms on programmable network switches?. Only FG-AQM and iRED have provided comparable performance statistics, so they are the only methods described in this section.

6.1 Evaluation of FG-AQM

In the paper about FG-AQM, Chen et al. [22] evaluated FG-AQM on its ability to handle microburst situations. They did this by conducting a test where normal background traffic (innocent flow) was sent over a duration of 250 seconds. They then picked three timestamps where they sent a numerous amount of burst flows and these affected normal (innocent) transmission flows. Figure 9(a) demonstrates that transmitting a high number of flows within a short timeframe without an AQM algorithm (No-AQM) significantly impacts the transmission flows, sometimes resulting in a throughput close to 0. This indicates that No-AQM is ineffective in handling microburst scenarios. The remaining figures in Figure 9(b)-(d) show that P4-RED, P4-PI2, and FG-AQM reduce the impact on the innocent flows to 73%, 61%, and 39%, respectively. Figure 9 shows a comparison of the loss rate in the three microburst scenarios. FG-AQM is reported to have a throughput 22% higher than P4-PI2 and 34% higher than P4-RED on average.

6.2 Evaluation of iRED

The researchers that implemented iRED also did an experiment comparing their algorithm for various buffer sizes, as well as comparing it to CoDel and PI2 [19]. Figure 10 shows the average RTT duration in millisecond (ms). Lower is better here, so iRED in the best-case scenario outperforms PI2 by 2.48x. This comes with a small footnote, because the buffer size of that iRED is only 64, and according to Table 2 0.038% of all packets are dropped. Compare this to the drop rate of CoDEL-P4 and PI2-P4, which were 0.005%, and it becomes clear that this difference is not as big as it seems. The researchers report that iRED 64 (best case) outperforms PI2 only by 0.74x in throughput. iRED drops packets in the ingress pipeline, while the CoDel and Pi2 drop packets in the egress pipeline. Notably, dropping packets in the egress pipeline results in packets traversing the entire switch pipeline

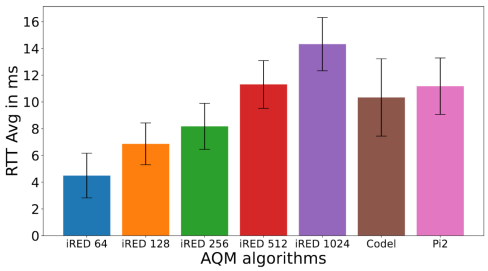


Fig. 10: Average RTT as reported by [19].

before being dropped, thereby wasting internal bandwidth. This difference in drop location may contribute to the observed variation in performance between these AQM algorithms.

7 Conclusion

The need for having AQM is clear, but integrating them on ASIC switches that large companies need has always been a problem due to its specific, complex, and inflexible design. With the introduction of P4 and the Intel Tofino, we are able to run the software directly on the switches. We have seen several attempts at porting AQM algorithms to the P4 language, such as CoDel, iRED, (Dual)PI2, PIE, and FG-AQM. However, all of these algorithms have drawbacks that arose from architectural constraints in this ecosystem. Most of these constraints could be solved by taking approximations, or workarounds, this however meant that the algorithms sometimes deviated from RFC standards.

Most notably, iRED and FG-AQM both seem to be highly performant AQM algorithms in their own right. Not using an AQM algorithm can lead to bad performance, such as a throughput nearing 0 in bursty situations, thereby underscoring the importance of implementing effective AQM algorithms. Lastly, iRED is a promising algorithm which is able to beat PI2 by 0.74x in throughput.

7.1 Future work

We suggest, just like some of the papers we have reviewed, that research should go into designing algorithms that work with the constrained architecture of programmable ASICs. It would be better to design a new algorithm that makes the best use of these limited fea-

IRED evaluated	Recirculated packets (%)	Bandwidth consumed
iRED 64	0.038	152Mbps
iRED 128	0.022	88Mbps
iRED 256	0.016	64Mbps
iRED 512	0.005	20Mbps
iRED 1024	0.005	20Mbps

Table 2: Table showing the recirculated packets, as reported by [19].

tures, instead of trying to fit existing AQM algorithms by constraining them.

One thing we have to mention is that the department of Intel that has made the P4-programmable Tofino series switch, have discontinued their development as of January 2023 [23]. The Tofino switches are the only ASICs, that we know of, to be programmable and run P4. The P4 language has been used by most of our papers to implement AQM algorithms, and the Intel Tofino to test them. This could imply that there is less of a need for developing better AQM algorithms in the P4 language, since there is not going to be any development going into the Tofino ASIC. What this means for future research on programmable network switches as a whole is unclear to us, since the announcement by Intel was very recent as of writing this paper and due to the lack of information on the topic.

One outcome that would benefit this field is a new programmable ASIC from a different, more established vendor of switches. If a state-of-the-art switch would be programmable, the adoption rate of them could be higher, leading to more possibilities in the future.

Besides the P4, there also exists other languages, such as the Network Programming Language (NPL) [24] and Domino [13]. Applications written in NPL can primarily be deployed on programmable ASIC switches from Broadcom [25]. Future research should also focus on languages such as these, which can be run on different types of switches.

7.2 Threats to validity

There are several potential threats to the validity of this study. Firstly, time constraints may have limited the scope of the rapid review, resulting in the potential exclusion of relevant papers. Additionally, author bias may have influenced the presentation of advantages and challenges for RQ1, potentially skewing perceptions of certain AQM algorithms. Finally, publication bias may have impacted the selection of papers for the review, potentially limiting the generalizability of our findings.

Acknowledgements

The authors wish to thank expert reviewer Saad Saleh for providing the topic, the initial selection of papers and reviewing the manuscript. The authors also would like to thank Germán Calcedo and Nikhita Prabhakar for reviewing the initial manuscript.

References

[1] Arielle Sumits. The History and Future of Internet Traffic. https://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic, 2015. Accessed on 21 March 2023.

[2] M. Allman, V. Paxson, and E. Blanton. RFC 5681: TCP Congestion Control, 2009.

[3] Intel. Intel® Tofino™. https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch/tofino-series.html, 2021. [Online; accessed 24-02-2023].

[4] Philippos Papaphilippou, Jiuxi Meng, and Wayne Luk. High-Performance FPGA Network Switch Architecture. In *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '20, page 76–85, New York, NY, USA, 2020. Association for Computing Machinery.

[5] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 8 1993.

[6] Kathleen Nichols, Van Jacobson, Andrew McGregor, and Jana Iyengar. Controlled Delay Active Queue Management. RFC 8289, 1 2018.

[7] Rong Pan, Preethi Natarajan, Fred Baker, and Greg White. Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem. RFC 8033, February 2017.

[8] Koen Schepper, Olga Albisser, Ing Tsang, and Bob Briscoe. PI2: A Linearized AQM for both Classic and Scalable TCP. In *Proc. Netdev 0x13*, 12 2016.

[9] Olga Albisser, Koen De Schepper, Bob Briscoe, Olivier Tilmans, and Henrik Steen. DUALPI2 - Low Latency, Low Loss and Scalable (L4S) AQM. In *Proc. Netdev 0x13*, 3 2019.

[10] Van Jacobson, Kathleen M. Nichols, and Kedarnath Poduri. RED in a Different Light. 1999.

[11] Wu-chang Feng, Kang G. Shin, Dilip D. Kandlur, and Debanjan Saha. The BLUE Active Queue Management Algorithms. *IEEE/ACM Trans. Netw.*, 10(4):513–528, aug 2002.

[12] Gergő Gombos, Maurice Mouw, Sándor Laki, Chrysa Papagianni, and Koen De Schepper. active queue management on the tofino programmable switch: The (dual)PI2 case. In *ICC 2022 - IEEE International Conference on Communications*.

[13] Anirudh Sivaraman, Mihai Budiu, Alvin Cheung, Changhoon Kim, Steve Licking, George Varghese, Hari Balakrishnan, Mohammad Alizadeh, and Nick McKeown. Packet Transactions: A Programming Model for Data-Plane Algorithms at Hardware Speed. *CoRR*, abs/1512.05023, 2015.

[14] The P4 Language Consortium. P4-14 language specification. https://p4.org/p4-spec/p4-14/v1.0.5/tex/p4.pdf, 2018.

[15] The P4 Language Consortium. P4-16 language specification. https://p4.org/p4-spec/docs/P4-16-v1.2.2.pdf, 2021.

[16] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. P4: Programming Protocol-Independent Packet Processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95, jul 2014.

[17] Ike Kunze, Moritz Gunz, David Saam, Klaus Wehrle, and Jan Rüth. Tofino + P4: A Strong Compound for AQM on High-Speed Networks? In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 72–80. Springer, 4 2021.

[18] Ralf Kundel, Jeremias Blendin, Tobias Viernickel, Boris Koldehofe, and Ralf Steinmetz. P4-CoDel: Active Queue Management in Programmable Data Planes. In *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–4, 11 2018.

[19] Leandro C. de Almeida, Guilherme Matos, Rafael Pasquini, Chrysa Papagianni, and Fábio L. Verdi. IRED: Improving the DASH QoS by Dropping Packets in Programmable Data Planes. In *Proceedings of the 18th International Conference on Network and Service Management*, CNSM '22, Laxenburg, AUT, 2023. International Federation for Information Processing.

[20] Mai Qiao and Deyun Gao. Fine-Grained Active Queue Management in the Data Plane with P4. In *2022 7th International Conference on Computer and Communication Systems (ICCCS)*, pages 174–179, 4 2022.

[21] Richelle Adams. Active Queue Management: A Survey. *IEEE Communications Surveys & Tutorials*, 15(3):1425–1476, 3 2013.

[22] Xiaoqi Chen, Shir Landau Feibish, Yaron Koral, Jennifer Rexford, Ori Rottenstreich, Steven A Monetti, and Tzuu-Yi Wang. Fine-Grained Queue Measurement in the Data Plane. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, CoNEXT '19, page 15–29, New York, NY, USA, 2019. Association for Computing Machinery.

[23] Tom's Hardware. Intel Sunsets Network Switch Biz, Kills RISC-V Pathfinder Program. Accessed on 21 Mar 2023.

[24] Network Programming Language. Open, High-Level language for developing feature-rich solutions for programmable networking platforms. Accessed on 21 Mar 2023.

[25] Broadcom. Trident4-X11C / BCM56890 Series. Accessed on 21 Mar 2023.

State of the Art: Securing broker-less publish and subscribe networks

Krishan Jokhan, Marten Struijk

Abstract— The publish-subscribe pattern is a popular way to realise loosely coupled message distribution networks. Throughout the years, different implementations have been proposed, each with their strengths and weaknesses. This paper focuses on the security aspect of broker-less pub-sub implementations by means of examining confidentiality, the protection of identities of a network participants, and authenticity and integrity. Additionally, it is interesting to see how well applicable these implementations are on the real world, as well as how performing and scalable they have been found.
It is not an easy task to compare different implementations based on the papers selected, due to the differences in reporting. Nevertheless, an overview was made on these proposals, based on the aforementioned topics.
If full subscription confidentiality is desired, it might be better to look into an implementation of publish/subscribe that uses a broker to transfer messages.

Index Terms—publish subscribe, software-defined networks, distributed systems, security, privacy, review

1 Introduction

With the rise of the Internet of Things and decoupled distributed systems the publish and subscribe pattern is gaining popularity. The publish and subscriber (pub-sub) pattern is nothing new in the field of computer science. It's a proven an useful technique to distribute the load over a network, or to give services the relevant information that they need to work properly. In traditional applications, these services often employ a broker that connects the publishers and subscribers and handles the routing, security and authorisation. Examples of these programs are RabbitMQ and Apache Kafka. The publish and subscribe pattern with a centralized broker is not suitable for all applications where pub-sub could be useful.

For this reason, an alternative exists. A brokers-less pub-sub system. These systems don't deploy a central broker that handles all requests. Rather it can be seen as a network of nodes that coordinates with each other to handle all relevant tasks. These networks are especially useful in the Internet of Things where no centralized place might exist. An example that deploys this strategy is a water quality monitoring system. In this study, the newly proposed solution seems even more efficient than the centralized approach [10]. This indicates that these methods might become more widely used in the future.

Securing these networks, and choosing the right type of security protocol can be crucial when defining these kinds of networks. For example: publishers and subscribers should be authorized to publish or subscribe such that bad actors cannot join the network. Additionally, messages/payloads need to be encrypted in such a way that they cannot be intercepted or faked. Other security issues also arise. When deployed in critical infrastructure these threads should be taken into account.

In order to solve these issues, different kind of access authentication protocols can be used which all have unique benefits and drawbacks.

By means of comparing different solutions, with this paper we hope to answer the following research questions:

1. What are the different kinds of security algorithms that exist for broker-less pub-sub systems?

2. What are their unique benefits and drawbacks?

- *Krishan P. Jokhan is with University of Groningen, E-mail: k.p.jokhan@student.rug.nl.*
- *Marten M. Struijk is with University of Groningen, Inc., E-mail: m.m.struijk@student.rug.nl.*

The remainder of this paper is structured as follows. We start with Section 2 in which we will cover some relevant background information and related works. After that we outline our approach in Section 3. We highlight how we found the papers and why they are chosen. Additionally we introduce the features on which we compare the algorithms. The results of this approach are outlined in Section 4 in which we cover each feature separately. In Section 5 the research questions are answered individually. After that the conclusion of this research follows in Section 6. Finally, idea's for future works are shared in Section 7.

2 Background

In this section we will provide some more information about some key topics that are discussed in this paper.

2.1 Publish and subscribe pattern

The publish/subscribe pattern is a way to transfer messages (or 'events') from a sender (the 'publisher') to a receiver (the 'subscriber'). When a publisher joins the pub-sub network, it states which type of events it will be sending by means of sending an *advertisement*. If a subscriber joins the network, it states what kind of events it wants to receive. As stated before, the transfer of messages from publishers to subscribers can be done via a central broker or in an interconnected manner between members of the network. In both situations, this is mainly done in two manners:

1. *topic-based*: a subscriber states what kind of topic (stored in a topic field of an event) it wants to receive. When an event is disseminated over the network, this topic field needs to be exactly matched in order for the subscriber to receive it [3].

2. *content-based*: a subscriber receives based on the entire content of an event [3]. Usually, a subscriber states which kind of *attributes* it is interested in, which then could be (partially) matched in order to be received.

The pub-sub pattern is loosely-coupled, which means that publishers and subscribers, inherently do not know about each other. A subscriber does not need to know *exactly* who will publish desired events, and vise-versa, the publisher does not need to know who desires the events published. Additionally, events are transmitted in an asynchronous manner, and members of the network do not need to be connected at the time an event is transmitted. These three properties make the pattern a suitable solution for message transmission in distributed systems. This is strengthened when using a broker-less pattern, removing a central point of the network.

2.2 Software-defined networking

Software-defined networking (SDN) is a counterpart to the traditional, static configuration of networks. It is a cloud-based network, allowing programmable configuration and easy management of the network [6]. The main power of SDN lays in the decoupling of network logic (control plane) and the forwarding of packets (data plane). Control operations are done by a *controller*, which contains and modifies the topology of the network. Usage of OpenFlow [2] allows for standardization for implementing SDN.

2.3 Related works

A search for related works that also compared the security of broker-less pub/sub networks didn't yield many results. A paper written in 2016 does highlight some of the algorithms, however a comparison is never made between them [9]. Another study in 2016 highlights a lot of different pub/sub mechanics but it includes broker and broker-less solutions. This gives the paper a very big scope [8].

3 Approach

In this section, we will describe the process by which we performed our research. This consists of the paper selection and ways of comparing the found solutions.

3.1 Paper selection and search

To get an overview on the state of the art regarding secure pub-sub systems, the following query was used on *Google Scholar* to collect papers proposing an implementation to the publish subscribe pattern:

(secure OR security) publish subscribe broker-less

For a paper to be included in this paper the following criteria were followed:

1. The paper has to describe an algorithm to implement broker-less pub/sub
2. Security of the algorithm must be highlighted.

From the papers found on Google Scholar, 22 papers were selected of which only 2 were considered relevant according to the criteria [7, 11]. Additionally, two papers given by the supervisor matched the criteria as well and were included [4, 5]. Finally, we incorporated a recent solution provided in the the Master's thesis of Braams [12]. Thus, in total, **5 papers were examined**. Summarising, these are namely the following:

- "Securing broker-less publish/subscribe systems using identity-based encryption" by Tariq et al. [4]
- "Securing broker-less publisher/subscriber systems using cryptographic technique" by Shitole et al. [11]
- "PLEROMA: A SDN-based high performance publish/subscribe middleware" by Tariq et al. [5]
- "Securing Publish/Subscribe systems using Software Defined Networks" by Braams [12]
- "AnonPubSub: Anonymous publish-subscribe overlays" by Daubert et al. [7]

3.2 Comparison

The provided solutions from the papers were compared with each other by means of looking at five different characteristics. The reason for comparing different solutions based on characteristics was chosen with the aim to provide a sort of common ground of comparison. Better comparison techniques, such as actually executing and measuring the different solutions, were unfortunately not feasible given the time available for this project.

As a basis, we took the paper of [4], to form characteristics to compare with other implementations. This paper is a well cited reference, and cited by the other papers as well. Two characteristics were decided upon based on the direction of this paper: security. These are as follows:

- *Confidentiality*: Does the implementation protect the identities of publishers and subscribers?
- *Authenticity*: Does the implementation check whether an event is authentic (e.g. not spoofed)?

We found practicability of an implementation important to discuss as well. Especially given the context in which publish/subscribe systems might be used, we focused on the decoupling property and overhead. We included two main characteristics to examine regarding this point of of view:

- *Applicability*: How well is the implementation applicable to existing systems? What needs to be changed in order to apply this implementation? Does it make use of any existing protocols?
- *Performance*:
  - *Overhead*: What is the overhead on e.g. joining/leaving publishers and subscribers?
  - *Scalability*: Does the implementation scale well w.r.t. growing publishers/subscribers and events?

Each paper will be examined individually based on each characteristic. Then, the insights on how the characteristics have been provided will be discussed in Section 5. Based on this, we conclude on what type of solution is best.

4 Results

In this section, we will discuss the results of our literature survey. Each subsection will cover an aspect as described in Section 3.2. Each characteristic will be examined in separate sections, but first, we introduce each selected paper with their given pub-sub solution.

One of the papers provided by the superviser proposes identity-based encryption to implement secure pub-sub systems. In this proposal Tariq et al. proposes that the attributes are the identities via which encryption keys are generated, therefore sometimes called *attrbute-based encryption* [4]. A subscriber is allowed to decrypt a message in case it has the right *credentials*, i.e. it needs to be subscribed to the attributes a message holds. The approach makes use of a central keyserver which generates keys per attribute, per subscriber. Furthermore, public keys are fairly simple to obtain, as any string can be used as the public key of a network participant.

Contrary to identity-based encryption, Shitole et al. use ellipticcurve cryptography to handle security [11]. Here, the goal is to use keys of small size, to enhance time and space efficiency. To further improve efficiency, cloud-services are used for computational tasks such as verification. Otherwise, the approach is similar, with a central keyserver handing out keys, with each attribute having a separate key;

To handle the dynamic nature of pub-sub systems, with (un)subscriptions and (un)advertisements happening while the system is running, Tariq et al. propose an SDN-based middleware called PLEROMA [5]. In essence, the SDN controller knows the entire topology of the network, and handles transmission of messages with high performance.

The Master's thesis of Braams also focuses on an SDN based solution for secure pub-sub systems [12]. Here, an efficient solution is proposed and implemented by means of SDN and *Programming Protocol-independent Packet Processors*, or *P4 switches* for short. Additionally, attribute-based encryption is used, namely *Cipertext attribute-based encryption* (CP-ABE).

Specifically aimed on achieving confidentiality, Daubert et al. propose *AnonPubSub*, a distributed solution to both protect the information transferred in the system, as well as the identities of both publishers and subscribers [7].

Comparing the algorithms directly is a rather hard task since no pre-made platform exists to perform this task.

4.1 Applicability

It is interesting to know what needs to be done or changed in an existing system for a solution to be able to be used in a real setting. Topics that surround applicability include ease-of-use, usage of existing protocols and interoperability with different pub-sub implementations.

The usage of identity-based encryption in contrast to a public key interface (PKI) reduces the amount of keys that need to be managed [4]. In this solution, any string is a valid public key, which makes it trivial to get the public key of a fellow participant. Furthermore, the usage of a single master public key makes it possible for smart-cards to be used to store the key.

A stregnth in applicability is the usage of existing protocols, in contrast to proposing novel ideas. AnonPubSub [7] also uses existing protocols, namely SCAMP for maintaining the neighbourhood and UDP for messaging. A strength in PLEROMA [5] lays in the usage of the IP protocol in order to encode attributes and transmit these. Furthermore, a controller creates virtual hosts to communicate with outside networks, and it does not need to know anything about the structure of these. This makes the algorithm applicable in a heterogeneous setting: together with systems that do not run the same algorithm.

The usage of cloud services reduce the cost of deploying and using the approach proposed in [11].

4.2 Performance

Performance is an essential factor when designing a publish/subscribe network. In this section, we will outline some of the performance characteristics of broker-less pub/sub-networks.

4.2.1 Overhead

Each of the papers viewed uses encryption to securely transmit data to the other nodes present in the network. For this paper, we will mostly look at the overhead of the encryption of the messages. Encryption can happen based on multiple factors. Either all messages received by the node are to be encrypted, or only some based on its attributes.

Making a fair comparison between all the different techniques is hard, because most papers didn't use the same hardware specifications nor, the same amount of nodes, attributes of message length. In general it can be said that the stronger the encryption, the more overhead there will be.

For example in the paper by Tariq, overhead is measured by comparing it to a less secure system (overhead 250ms) [4]. Additionally they provide the CPU usage for all steps. The paper by Braams also comparse the overhead of using different filters for content types. It notes that for it's implementation OR filters have a much higher decryption time [12]. The paper about the PLEROMA algorithm mostly looks at the overhead of the controller which which is less relevant for this section [5].

The paper by Shitole [11] doesn't show any evaluation of the overhead of the system.

4.2.2 Scalability

When talking about scalability the metrics most often used are the events that are published using a network of a large number of nodes. Scalability is important if the network consists of a large number of nodes that need to effectively communicate the events with minimal or no loss of data.

In broker-less pub-sub systems nodes are connected to several other nodes. Nodes can receive messages and have the option to either decrypt its message contents, and/or forward it to other connected nodes. Part of the security protocols is building these networks and ensuring they receive the right encryption keys.

Regarding the sharing of encryption keys, solutions that use a single master server tend to be the best in terms of performance. For example, a master server key server can easily be replicated. These key servers can then be used to provide the nodes with the encryption and decryption keys [4]. A downside to this approach is that a new centralized place has been introduced, in a system that tries to eliminate the need for such a centralized entity.

Central key server	[4], [5],[7], [12], [11]
Other solutions	

Table 1. Key distribution method

As can be seen in Table 1 all solutions make use of an centralized key server. This means that in terms of scalablity there won't be much difference in this aspect.

In terms of routing different solutions are used. The configuration of the network can either be handled by the nodes themselves or a master server that has knowledge of all nodes and routes. When a master server handles the routing configuration it's important the algorithm is efficient and doesn't overload certain nodes. Additionally it's important that a message isn't forwarded many times without actually being consumed.

The system described by Tariq [4] uses a tree structure to which a request can be made to connect. The node will then be connected to the tree in such a way that their attributes are contained by their parent nodes. This ensures that the tree that events are send only to nodes that either need to forward them or consume them. Scaling this solution for big trees can be an issue since the connection message needs to travel.

AnonPubSub also uses the nodes to configure the network. It uses SCAMP, a peer-to-peer lightweight membership service for large scale communication [7] [1]. The use of SCAMP ensures that no node has knowledge over the whole network and it's particpants. Networks loops are detected by the algorithm in AnonPubSub. Due to not knowing the full extend of the network size, it could be that a message travels quite some nodes, while a shorter route might be possible.

PLEROMA uses a different strategy and opts to use a central routing server that has knowledge of all routes. Using this knowledge it can construct multiple spanning trees. This ensures that no node is flooded with messages and that the maximum amount of hops between nodes can be minimized.

Central routing server	[5], [12]
Handled by nodes	[4], [7]
Not explicitly mention	[11]

Table 2. Routing mesh

Generally the central routing server solutions scale better because less messages are required to reconfigure the network when a node connects or disconnects from a certain topic of attribute.

4.3 Confidentiality

An important goal to achieve security is to achieve confidentiality. The goal is to hide the identities of publishers and subscribers in the network, which by definition is already present to some extent: publishers do not know who will receive their messages and vise-versa. This is covered in all solutions discussed in this paper.

However, when utilizing a broker-less variant of the pub-sub network, it is hard to achieve full confidentiality on all fronts. For instance, [4] states that *full subscription confidentiality* cannot be reached when subscribers are clustered together based on the attributes they are subscribed to. This automatically makes it explicitly hard to achieve confidentiality for a number of the examined solutions [11, 5, 4].

Instead, for the solution of [4], only a weaker variant of subscription confidentiality is realised via multi-credential routing. This makes sure that at most information about neighbouring subscribers is known. Additionally, the credential generation proposed allows for splitting a credential into more fine-grained variants. If each of these are spread across different parents, it is harder to recover the initial credentials. This does not make the solution watertight, the different parents could

still collude in order to undo the mitigation. To provide event confidentiality, subscription keys should always be bound together per subscriber, as they could separately (or subsets of them) be part of different subscribers.

Similarly, Multi-credential routing is also used in [11] to achieve subscription confidentiality. This technique is extended to also protect publishers. Furthermore, it was found that the usage of the cloud made it possible to intercept man-in-the-middle attacks. This makes sure that, for instance, eavesdropping is harder to accomplish. The paper admits that the usage of ECC might not be the strongest solution for encryption, albeit time and space efficient, and suggests that an attribute-based variant might be better.

Similarly, the PLEROMA algorithm [5] makes use of a similar attribute mechanism, and makes sure false positives are minimized, by making sure coverage rules are met: a publisher and subscriber must match a subspace in a tree in order for an event to be forwarded. Furthermore, only the direct switch connected to the host a participant (publisher or subscriber) is running on knows of its identity. Each controller in the SDN network only knows of its neighbouring networks, and the identity of neighbouring *controllers* stays unexposed.

The AnonPubSub solution provides notification and subscription confidentiality. Pseudonyms are sued in order to hide the content of an attribute, and encryption is used to transfer notification. A key is only received by members of closed attribute groups.

Encryption is a recurring tactic to provide confidentiality. Both an asymmetric public/private key as well as a symmetric key solution are used in [4] in order to find a balance in efficiency and strength: symmetric keys are used to encrypt messages as they are not dependent on the plaintext content. The solution provided in [12] makes sure all data transmitted is encrypted, such that switches are not able to eavesdrop and access data. Furthermore, subscribers are independent of each other, and are unable to collude in order to view content.

4.4 Authenticity

Authenticity makes sure that messages are *authentic*, for instance, coming from trusted sources and participants of the network. As well, following [4], subscribers should only be able to receive events they are allowed to. In this way, it makes it related to confidentiality, and some of the tactics summarised there are applicable on this topic as well.

All the papers discussed in this paper have points to provide authenticity. The identity-based solution of [4] verifies events by means of the master public key and a defined relation between different parts of the encryption, as well as the usage of signatures. The latter is also used to enforce integrity and authenticity in AnonPubSub [7]. Here, signatures are used to authenticate publisher advertisements and notification messages. This is additionally used to prevent forms of spamming.

Authenticity is also handled in [12] to make sure only publishers that are authenticated are allowed to send messages. Finally, the usage of a third-party authority attempts data tampering in the cryptographic solution provided in [11]. In this design, all verification is done by the third party in order to minimize cost on the user-side.

5 Discussion

In this section, we will discuss the results of our short study and its possible shortcomings. We try to answer the research questions below.

As it turns out comparing algorithms is a hard task. This is because each of them has been created for a specific context, and even if this context is the same the testing methods have not been standardised such that papers can easily be compared with each other. This makes it hard to judge the progress in the field, but also the effectiveness of new works. In Section 7 we will touch on this a bit more.

5.1 What are the different kinds of security algorithms that exist for broker-less pub-sub systems?

Each of the approaches have been discussed in Section 4. We found that there exists a similarity in the solutions examined. Since the papers all propose a broker-less pub-sub implementation, participants of the system are ordered in trees, which are used to relay events.

The approaches mainly differ in how encryption is realised and on what fronts. Furthermore, some approaches make use of third parties such as cloud-services or a third party authority for key management [11].

5.2 What are their unique benefits and drawbacks?

A big drawback therefore lays in subscription confidentiality, it is impossible to achieve full subscription confidentiality: neighbouring subscribers (i.e. with similar subscriptions) will know of each other by means of coverage (credentials to be more coarse or fine-grained) as this is how messages are delivered to appropriate subscribers. This was explicitly stated in three of the papers [11, 5, 4], but one may suggest that this applies to the remainders as well. The problem is somewhat mitigated by providing a weaker form of subscription confidentiality.

As seen in the results, there is a great benefit in using existing protocols and technologies. Using SDN as a basis for networking such as in PLEROMA [5] allows another layer of decoupling, which is ideal for distributed systems and cloud computing.

While cloud services provide a relaxation in costs [11], one might question whether these services are ideal, especially in highly sensitive environments where third parties are lesser trusted. This can be extended to the usage of third-party authorities, also used in [11]. Furthermore, the paper admits that ECC is not the strongest encryption method out there, all of this making the proposal possibly less suitable for highly sensitive environments. Because multiple papers aim on an attribute-based encryption, this might be a good property for a secure pub-sub system.

All the proposed solutions aim to protect authenticity in different manners.

In terms of performance all proposed solutions we found used a central key server to distribute encryption keys and certificates. One could argue that these systems are less secure if this server get's compromised and can be seen as a single point of failure. This is a tradeoff in terms of scalability since a single master key server can more easily be replicated. And it prevents keys from being send over the complete network each time a subscriber or publisher joins.

Overhead wasn't something that was comparable between the papers in a significant way due to the different approaches in terms of measuring this, or even just completely not mentioning it.

Another finding that was rather unexpected was that all solutions deployed some kind of centralized key management system. No real difference exists between the different solutions in this regard. This is rather surprising since by removing the broker, one would expect no centralized entity to exist. However as it turns out, it is hard to keep the security of a system high if there is no solid key management. One can conclude that the main focus for all algorithms is to ensure there is no centralized place for message traffic, but for key management they all conclude that it's fine to do this centralized. We expand more on this in Section 7.2.

6 Conclusion

In this paper, five implementations of the publish/subscribe pattern were examined and discussed. On a total of five main characteristics, the approaches were discussed based on applicability, performance, scalability, confidentiality and authenticity. We found that there exists a similarity in some approaches, mainly in the organisation of members of the network by means of using spanning trees.

If full subscription confidentiality is desired, it might be better to look into an implementation of publish/subscribe that uses a broker to transfer messages, in contrast to categorizing subscribers in trees. Broker-less implementations of the publish/subscribe pattern might not be optimal for all situations, such as highly sensitive environments, but it seems that for many, a good compromise can be made.

7 Future works

In this section we will outline some idea's for future works we think are interesting to explore based on this paper.

7.1 Standardized testing framework

A lot of different implementations exist but no normalized effective means to compare them seem to be in place. This makes it hard for reviewers, and compare new algorithms. A future work could focus on building or defining a standardized method of testing such that algorithms can easily be compared in terms of performance, and maybe even security.

7.2 Broker less pub/sub network without centralized key server

An interesting finding of this paper is that all solutions deploy a key server. This centralized key server handles encryption keys. Although this simplifies the process of key management, once could argue that this goes against idea of having a centralized entity in the network. An interesting research future would could focus on maintaining security without such a key server, such that a completely decentralized network can be created.

ACKNOWLEDGEMENTS

The authors wish to thank Boris Koldehofe for being an expert reviewer and having a meeting with us.

REFERENCES

[1] A.J. Ganesh, A.-M. Kermarrec, and L. Massoulie. "Peer-to-peer membership management for gossip-based protocols". In: *IEEE Transactions on Computers* 52.2 (2003), pp. 139–149. DOI: 10.1109/TC.2003.1176982.

[2] Nick McKeown et al. "OpenFlow: Enabling Innovation in Campus Networks". In: *SIGCOMM Comput. Commun. Rev.* 38.2 (Mar. 2008), pp. 69–74. ISSN: 0146-4833. DOI: 10.1145/1355734.1355746. URL: https://doi-org.proxy-ub.rug.nl/10.1145/1355734.1355746.

[3] Sasu Tarkoma. *Publish/subscribe systems: design and principles*. John Wiley & Sons, 2012.

[4] Muhammad Adnan Tariq, Boris Koldehofe, and Kurt Rothermel. "Securing broker-less publish/subscribe systems using identity-based encryption". In: *IEEE Transactions on Parallel and Distributed Systems* 25 (2 Feb. 2014), pp. 518–528. ISSN: 10459219. DOI: 10.1109/TPDS.2013.256.

[5] Muhammad Adnan Tariq et al. "PLEROMA: A SDN-based high performance publish/subscribe middleware". In: Association for Computing Machinery, Dec. 2014, pp. 217–228. ISBN: 9781450327855. DOI: 10.1145/2663165.2663338.

[6] Kamal Benzekki, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. "Software-defined networking (SDN): a survey". In: *Security and Communication Networks* 9.18 (2016), pp. 5803–5833. DOI: https://doi.org/10.1002/sec.1737. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1737. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1737.

[7] Jörg Daubert et al. "AnonPubSub: Anonymous publish-subscribe overlays". In: *Computer Communications* 76 (2016), pp. 42–53.

[8] Emanuel Onica et al. "Confidentiality-Preserving Publish/Subscribe: A Survey". In: *ACM Comput. Surv.* 49.2 (June 2016). ISSN: 0360-0300. DOI: 10.1145/2940296. URL: https://doi-org.proxy-ub.rug.nl/10.1145/2940296.

[9] Ankita V. Terkhedkar and Medha A. Shah. "Approaches to provide security in publish/subscribe systems — A review". In: *2016 International Conference on Inventive Computation Technologies (ICICT)*. Vol. 1. 2016, pp. 1–4. DOI: 10.1109/INVENTIVE.2016.7823221.

[10] Alif Akbar Pranata, Jae Min Lee, and Dong Seong Kim. "Towards an IoT-based water quality monitoring system with brokerless pub/sub architecture". In: *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. 2017, pp. 1–6. DOI: 10.1109/LANMAN.2017.7972166.

[11] Shilpa Shitole and A. D. Gujar. "Securing broker-less publisher/subscriber systems using cryptographic technique". In: Institute of Electrical and Electronics Engineers Inc., Feb. 2017. ISBN: 9781509032914. DOI: 10.1109/ICCUBEA.2016.7860073.

[12] C.S. Braams. "Securing Publish/Subscribe systems using Software Defined Networks". 2022. URL: https://fse.studenttheses.ub.rug.nl/29103/.

Logistic Regression and Linear Discriminant Analysis: A Comparative Overview and an Empirical Time-Complexity Analysis

Eelke Landsaat, Johanna Lipka

Abstract— Classification of data into groups based on predictor variables has been a major concern in many research areas for decades. While modern approaches can obtain high predictive performance, they often come with the drawbacks of high complexity and computation time. Considering this, classical approaches still play an important role in the field, and increasing practical understanding of their use cases may have a positive impact on newer approaches as well. Two such approaches, logistic regression (LR) and linear discriminant analysis (LDA), have been the subject of several comparisons in previous works.
In this paper, the methods and results of several such works are summarized and compared, taking into account various hyperparameters, evaluation metrics, and use case specifics. Additionally, an empirical time complexity analysis is carried out for the two model types to enrich the discussion of which to use with considerations regarding the training time. Strengths and limitations of the work are discussed.
Ultimately, a guideline is proposed outlining which model to employ given varying use case characteristics. The general conclusion is to use LDA only when training time is of the essence, as an LDA model is found to be trained 33% faster than an LR model. In other cases, LR tends to be the better option due to higher predictive performance, with few exceptions.

Index Terms—Linear discriminant analysis, logistic regression, multivariate statistics, classification

1 Introduction

The objective of this paper is to summarize and compare the results of Lei and Koehly [11], Liong and Foo [12], and Pohar, Blas, and Turk [15], to provide a comprehensive overview of the relative performance of linear discriminant analysis (LDA) and logistic regression (LR). We complement the results with experiments comparing LDA and LR in terms of their empirical time complexity. Ultimately, we propose a guideline indicating when to use which of these techniques and with which parameter settings.

LDA is a generalization of Fisher's linear discriminant [5] and is used to find a linear combination of features that separates objects into distinct classes. It is related to other statistical methods such as analysis of variance (ANOVA) [17], regression analysis [6], and principal component analysis (PCA) [1].

LR models the probability of an event taking place based on various predictor variables. The logistic model was first developed by statistician Joseph Berkson in the 1940s as a way to model the relationship between binary outcomes and predictor variables in medical research [4]. The logistic regression model was further refined by David Cox in the 1950s and 1960s, who introduced the concept of maximum likelihood estimation for fitting the model parameters [4]. The method became widely used in medical research and social sciences in the 1970s and has since become a standard tool in many fields, including epidemiology, marketing, finance, and machine learning.

While the predictive accuracy of LR and LDA have been examined in existing literature [11] [12] [15], there is still a need for a comprehensive overview and guidelines motivating the choice of method. Moreover, the existing comparisons rarely take computation time into account, even though this may be a deciding factor for some. This paper aims to close these knowledge gaps by summarizing the results of previous comparisons and supplementing them with an empirical time complexity analysis.

LR and LDA are explained in-depth in sections 2.1 and 2.2, respectively. Sections 2.3 and 2.4 describe the hyperparameters and evaluation metrics used by [11, 12, 15]. Section 3 outlines the methods and results obtained by [11, 12, 15]. In section 4, we present our contribution to the discussion in the form of an empirical time complexity analysis. In section 5, we aggregate and discuss our findings and those of [11, 12, 15]. Section 6 summarizes and concludes the discussion of the results. Section 7 enumerates some possible directions for future studies.

- *Eelke Landsaat is with the University of Groningen, E-mail: e.landsaat@student.rug.nl*
- *Johanna Lipka is with the University of Groningen, E-mail: j.lipka@student.rug.nl*

2 Preliminaries

LR and LDA are two multivariate statistical methods used to classify data into distinct groups. While they can both be used to create linear classification models, they differ in their preconditions and assumptions made about the data.

2.1 Logistic Regression

Logistic regression models are statistical tools that describe the relationship between a qualitative dependent variable and a set of independent or predictor variables [9]. They are used to investigate how the predictor variables impact categorical outcomes, such as the presence or absence of disease. When the outcome is binary, the model is referred to as a binary logistic model. When only one predictor variable is used, we speak of a simple logistic regression, whereas multiple predictors, such as risk factors and treatments, make the model a multivariate or multiple logistic regression [14]. Since all three of the papers we are summarizing and comparing make use of multivariate logistic regression with a binary outcome variable, this is the main focus of this section.

Multivariate Logistic Regression When more than one predictor variable is used, the model is referred to as a multivariate logistic regression. A set of p predictor variables may be denoted as $\mathbf{x}^T = (x_1, x_2, ..., x_p)$, and the probability that the outcome Y is present may be denoted as $P(Y = 1|\mathbf{x}) = \pi(\mathbf{x})$. The logit function for multivariate logistic regression is given by:

$$g(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p \quad (1)$$

This function produces a value that may range from negative to positive infinity that can be used in the logistic regression function. The parameters $\beta_0, \beta_1, ..., \beta_p$ are the coefficients that represent the effect of the input variables on the odds of the event happening. These may be referred to as a vector $\boldsymbol{\beta}^T = (\beta_0, \beta_1, ..., \beta_p)$. $\boldsymbol{\beta}$ is obtained by fitting the logistic regression model to the training data. Using Equation 1, the multivariate logistic regression may then be described as follows:

$$\pi(\mathbf{x}) = \frac{e^{g(\mathbf{x})}}{1+e^{g(\mathbf{x})}} \tag{2}$$

Fitting the Model Fitting the model to the data requires a set of n observations $(\mathbf{x_i}, y_i)$, with $i = 1,2,...,n$, and an estimation for $\boldsymbol{\beta}$ in Equation 1. The logistic regression model tries to find the best $\boldsymbol{\beta}$ that can predict the outcome variable (i.e., the probability of the event happening) based on the input variables. This is achieved using the maximum likelihood method, which tries to find values for the unknown elements of $\boldsymbol{\beta}$ that maximize the probability of obtaining the observed set of data [9]. First, a likelihood function is constructed as follows:

$$l(\boldsymbol{\beta}) = \prod_{i=1}^{n} \pi(\mathbf{x_i})^{y_i}[1-\pi(\mathbf{x_i})]^{1-y_i}. \tag{3}$$

Since it is more convenient for computations and stability, the log of Equation 3 is used. To optimize the parameters $\beta_0, \beta_1, ..., \beta_p$, the log likelihood function is differentiated with respect to $\beta_j, \forall j \in [0..p]$, and the resulting equations are set to 0 and solved for β_j. Solving for β_j requires iterative methods that adjust the values until a solution is found. For this reason, logistic regression is computationally expensive, especially with large sample sizes and many predictor variables.

In practice, there are multiple choices for these iterative methods, such as Gradient descent [16], the Newton-Raphson method [2], or L-BFGS [13]. The choice of method may depend on various factors, such as the number of samples, the number of input variables, and the structure of the data.

Theoretical Time Complexity In the following, we may refer to the sample size as n. Training an LR model involves n operations in $O(p)$, resulting in a total time complexity of:

$$O(np) \tag{4}$$

2.2 Linear Discriminant Analysis

As opposed to Logistic Regression, where no assumptions are made on the distribution of the independent variables, LDA has been developed for normally distributed independent variables. It is used to find a linear combination of independent variables that maximally separates different classes. LDA makes the assumption that classes have common covariance matrices (see also subsection 2.3) [8]. These assumptions simplify the calculation of the posterior probabilities and allow LDA to be implemented efficiently [10].

The model Since Lei and Koehly [11], Liong and Foo [12], and Pohar, Blas, and Turk [15] only consider binary outcome variables in their experiments, we focus on an LDA model with only two classes. The population may be denoted as P and is divided into k classes $\Pi_1, \Pi_2, ..., \Pi_k$. In P, each item can be classified into exactly one of the classes. The measurements of an item in P is denoted as $\mathbf{X} = (X_1, ..., X_r)^T$, where $X_1, X_2, ..., X_r$ are the discriminating or feature variables, which have been chosen to distinguish between the k classes [10].

LDA is based on Bayes' theorem, which calculates the posterior probability that $\mathbf{x}$ belongs to $\Pi_i, i = 1,2$:

$$p(\Pi_i|\mathbf{x}) = P(\mathbf{X} \in \Pi_i|\mathbf{X} = \mathbf{x}) = \frac{f_i(\mathbf{x})\pi_i}{f_1(\mathbf{x})\pi_1 + f_2(\mathbf{x})\pi_2}, \tag{5}$$

where π_i is the prior probability that the observation belongs to the class Π_i, and $f_i(\mathbf{x})$ is the conditional multivariate density of $\mathbf{X}$ in the class Π_i. An observation $\mathbf{x}$ is assigned to Π_1 if:

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > \frac{\pi_2}{\pi_1}. \tag{6}$$

LDA makes Bayes' rule classifier more specific by assuming that the multivariate probability densities are normally distributed and have a common covariance matrix. The linear discriminant function for LDA can be written as:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0, \tag{7}$$

where $\mathbf{x}$ is the input vector of features, $\mathbf{w}$ is a weight vector, and w_0 is a bias term. The weight vector and bias term are computed based on the mean and covariance matrix of each class in the training data (Least squares solution). Equation 7 is formally the same as Equation 1 but written differently, showing that the two methods LR and LDA do not differ in fucntional form but mainly in the estimation of coefficients [15].

Theoretical Time Complexity The time it takes to train an LDA model is in (see [3]):

$$O(ndt + t^3), \tag{8}$$

where $t = \min(d, n)$. Since we will only consider the case where $d < n$, we can simplify this time complexity to

$$O(nd^2) \tag{9}$$

for our purposes.

2.3 Hyperparameters

Lei and Koehly [11], Liong and Foo [12], and Pohar, Blas, and Turk [15] investigate LR and LDA with varying hyperparameters. These experiment variables, indicated with italics, are explained in this section.

The *prior probability* is a setting that LDA uses as an initial estimation of the group distributions. Groups have *covariance matrices* associated with them, which contain variances of predictor variables along the main diagonal and covariances between them in all other positions. The *cut-score* of a model in a binary setting is the output value below which samples are classified as one group and above which they are classified as the other. The *level of normality* of a data set is used as a hyperparameter as well, although not always clearly defined. It can be measured using statistical tests such as the Shapiro-Wilk test. The *Mahalonobis distance* is a measure of the distance between two groups of data points and defined as the distance between the group centers. Finally, the *direction of distance between group means* is used as a hyperparameter as well.

2.4 Evaluation Metrics

In this section, we outline the performance evaluation metrics used by [11, 12, 15] to examine the performance of LDA and LR. The evaluation metrics themselves are italicized.

The *classification error* is defined as the fraction of incorrectly classified objects. Conversely, the *accuracy* is the fraction of correctly classified objects and can be calculated by 1 - classification error.

The *B-index* can be used to describe the performance of model predictions. It is the complement of the more well-known Brier-score and can be calculated as:

$$B = 1 - \sum_{i=1}^{n}(P_i - Y_i)^2/n, \tag{10}$$

where n is the number of samples, Y_i is the actual group membership of sample i, and P_i is the classification probability of sample i into group 1. A B-index of 1 corresponds to perfect prediction, while 0 indicates perfect false prediction. The b-index provides a more nuanced picture than the accuracy, as it takes into account the predicted probability of group membership, as opposed to the predicted label only.

The *C-index* measures the ability to discriminate between classes. It is calculated as [15]:

$$C = \sum_{i=1}^{n}\sum_{j=1}^{n}[I(P_j > P_i) + \frac{1}{2}I(P_j = P_i)]/n_0 n_1, \tag{11}$$

where I is an indicator function, and n_0 and n_1 are the numbers of elements with predicted class labels 0 and 1, respectively. A C-index of 1 indicates perfect discrimination, while 0.5 corresponds to random prediction. Note that the C-index does not depend on the true classes of the samples.

The *Q-index* is another measure of predictive accuracy, but on the interval $[-1,1]$. It is calculated as [15]:

$$Q = \sum_{i=1}^{n} \left[1 + \log_2(P_i^{Y_i}(1-P_i)^{1-Y_i})\right]/n, \tag{12}$$

with the aforementioned variable definitions. A Q-index of 1 denotes perfect prediction, while 0 corresponds to random prediction. Further discussion of these indexes as evaluation metrics is given by Harrel and Lee [7].

3 Comparison Papers

In this section, we discuss the methods used by Lei and Koehly [11], Liong and Foo [12], and Pohar, Blas, and Turk [15] to compare the classification performance of LDA and LR under various conditions, as well as the results they obtained.

3.1 Methods

Pohar, Blas, and Turk [15] first consider the case where all assumptions for LDA are met and examine the influence of sample size, covariance matrix, Mahalanobis distance, and the direction of distance between the group means. They also compare the performance of both methods in regard to categorization and when the normality condition of LDA is not met (see Table 1).

Liong and Foo [12] examine seven real datasets with different degrees of normality, number of independent variables, and sample size. Moreover, LDA performance was also tested with different prior probabilities.

Lei and Koehly [11] use simulated data to examine the performance of LDA and LR with equal and unequal covariance matrices, a small and a large degree of group separation, different prior probabilities (50:50, 25:75, 10:90) and varying sample size (100 and 400). They used three predictor variables which were assumed to be normally distributed within groups.

Table 1: Hyperparameters per paper.

Lei and Koehly [11]	Liong and Foo [12]	Pohar, Blas, and Turk [15]
Sample size, Prior probability, Covariance matrices, Squared Mahalonobis distance, Cut-score	Sample size, Prior probability, Level of Normality, Number of predictor variables	Sample size, Covariance matrices, Level of Normality, Mahalanobis distance, Direction of distance between group means, Categorization

Pohar, Blas, and Turk [15] use the B, C, and Q indices to compare the predictive accuracy of LR and LDA. They state that while classification error is an insensitive and statistically insignificant measure, they still include it in some experiments since it is an intuitive measure. The C-index is a measure of discrimination between the groups, and the B and Q indices are measures of predictive accuracy.

Liong and Foo [12] compare and evaluate the performance of LDA and LR using the percentage of accurate classifications although admitting that it is insensitive and statistically insignificant. They also use the B-index as a measure of predictive accuracy.

Lei and Koehly [11] only consider the classification error when comparing the two methods.

Table 2: Evaluation metrics per paper.

Lei and Koehly [11]	Liong and Foo [12]	Pohar, Blas, and Turk [15]
Classification error	Accuracy, B-index	Classification error, B-index, C-index, Q-index

3.2 Results

The results of the experiments detailed in the previous section are discussed in this section.

Pohar, Blas, and Turk [15] found that in the case where the independent variables are normally distributed, the sample size has the largest impact on the difference between LDA and LR performance. As the training sample size is small, it can differ substantially from the test sample in terms of distribution, which leads to LR performing slightly worse in terms of the B, C, and Q indices and the classification error. As the sample size increases, the sampling distribution will increase in stability and therefore lead to better results for LR. They found that the LDA coefficient estimation also becomes more accurate as the sample size increases, but since the LR indices are increasing faster the difference between the models decreases as the sample size grows. They found that the differences between the methods become negligible with a sample size larger or equal to 50.

Moreover, they found that for the other parameters, e.g., the covariance matrix/ correlation, the direction of distance between the group means, and the Mahalanobis distance, LDA and LR do not differ much in terms of performance, though LDA is slightly better. Only when the Mahalanobis distance is large LR will outperform LDA.

When considering cases where the normality assumptions are not met, and the distributions are skewed, Pohar, Blas, and Turk [15] found that LR performs better than LDA as skewness grows. The sign of the skewness does not matter.

Liong and Foo [12] found that for all levels of normality in their study, LR outperformed LDA. Even when the normality assumption was met fully, LR resulted in a higher B-index than LDA. Moreover, they found that with increasing sample size, LR had a higher percentage of correct classification than LDA. Moreover, they compared the performance of LDA and LR in terms of computing time for three datasets of different sizes and found that overall, LR was faster. Finally, they examined the effect of prior probability on the performance of LDA and compared the B-index values using equal prior probability to those using computed prior probability. They found that the computed prior probability values lead to a higher B-index in two out of three cases.

Lei and Koehly [11] found that the optimal cut score is 0.5 for LDA with proportional or accurate prior probabilities or LR with a representative sample if the goal is to reduce the total misclassification error. When the separate group misclassification rate is important, e.g., in scenarios where it is more harmful to misclassify one group than another, they found that LR or LDA with extreme priors (10: 90) and a cut-off score of 0.5 is the best method for reducing the large-group classification error, and LDA with equal priors and a cut score of 0.1 for reducing the small group classification error. They do, however, advise the use of the optimal methods for separate group misclassification only with great caution and when the cost of misclassification is evidently greater for one group than the other.

Additionally, they found that LDA, performed on distributions with unequal covariance matrices, only performed worse in large-group misclassification. Contrary to their expectation, for the small-group misclassification rate almost all combinations of method and cut-score performed better when the covariance matrices were unequal.

4 Empirical Time Complexity Analysis

Although Lei and Koehly [11], Liong and Foo [12], and Pohar, Blas, and Turk [15] all thoroughly investigated the performance of LR com-

pared to LDA in terms of classification results, the time consumed by each algorithm to obtain these results played a small role only in [12]. In this section, we present an empirical time complexity analysis comparing the computation time for training LR and LDA. Our goal here is to refine the guidelines for which model to choose in a practical setting by varying the parameters, n and p, of the theoretical time complexity of LR and LDA and observing the effect on the time consumed by each algorithm. Specifically, we attempt to verify the time complexities at the chosen scales and determine which model is faster for which parameter values.

4.1 Method

For the sake of imitating a realistic machine learning environment, we used the `scikit-learn` package for Python as a provider of the LDA and LR training algorithms. For LDA, the `LinearDiscriminantAnalysis` class was used. For LR, we used the `LogisticRegression` class. All elements involving pseudo-random number generation were seeded with seed 0.

4.1.1 Models

Here, we describe the specifics of the models used for the time complexity analysis. For both LDA and LR, we fit a new instance of the respective model class for each experiment.

LDA The pivotal property affecting the time complexity of training a predictive model is the iterative method (solver algorithm) used. We opted for the singular value decomposition solver, as it is most efficient for data sets with a large number of samples (whenever we use a superlative term, such as 'most efficient', in this context, we imply a comparison with the other options available in the `scikit-learn` package). We let the algorithm base the prior distributions on the samples, which were always divided 50/50, to simulate an absence of knowledge about the population distributions by the user. Furthermore, we set the tolerance for considering a singular value as significant to its default value of 0.0001, not to make any assumptions about the specific use case. The parameters `shrinkage`, `n_components`, and `covariance_estimator` were all set to `None` to mimic a plain use case of LDA. We set the parameter `store_covariance` to `False`, not to induce any extra computation time which would only be beneficial when more predictions would be performed with varying solvers.

LR As a solver for LR, we used stochastic average gradient descent with averaging (SAGA), since it is most efficient for data sets with many samples and many features and it supports L1 penalty terms. To represent the average LR use case, we used a combination of L1 and L2 penalty terms with elastic net, with an `l1_ratio` of 0.5. For the same reason, we set the regularization strength, controlled by the `C` parameter, to the default value of 1. As a tolerance for the convergence stopping criteria, we used the default value of 0.0001, not to make any assumptions about the usage scenario. To imitate an absence of knowledge about the data, we set the `fit_intercept` parameter to `True`, which adds a bias term to the model. For the same reason, we set the `class_weight` to balanced, which bases the class weights on the distribution of the sample data. We set `random_state` to 0 to ensure reproducibility of the experiments. The iteration limit was set to 1000, as this was experimentally sufficient for the algorithm to converge in all experiments. Since the outcome variable is binary, we used the one-versus-rest strategy set by the `multi_class` parameter. The remaining parameters were all either not relevant in this context or not applicable in combination with the SAGA solver.

4.1.2 Data

To be able to obtain many uniformly distributed data points, we used simulated data. The data always consisted of 2 outcome variables, in line with the comparison papers, and a varying number of predictor variables. The predictor variable distributions per outcome variable were Gaussians and only differed in their mean. This is because it is generally not advisable to use LDA if approximate equality of the covariance matrices for each class has not been shown, so we assume that the user has investigated this. We set the mean vectors of the 2 distributions to have norm 0.5 and to point in opposite directions, putting them a unit distance apart around the origin. Following the assumption of both LR and LDA that predictor variables are not highly collinear, we set the covariance matrices to be diagonal matrices with uniformly distributed entries in the range $[0,1)$. We assume that the user made sure of this condition, either by concluding that it is approximately true for the data naturally or by making it true using some form of preprocessing, such as PCA.

Experiments were performed with numbers of predictor variables in the range $[10,20,...,160]$ and numbers of samples in the range $[10000,20000,...,200000]$. These ranges were chosen, as they induce similar changes in training time. For each combination of a number of features and a number of samples, 1 data set was generated and used for training 10 times. This is to allow for the derivation of medians and confidence intervals and to alleviate the effect of background processes on the reported execution time.

4.2 Results

As can be observed in Figure 1, the LDA training experiments resulted in a rather smooth 'landscape' of execution times. There seems to be somewhat equal variation in both dimensions, with a clear minimum and maximum in execution time at the minima and maxima of n and p, respectively. Figure 2 is highly similar to Figure 1, with the exception that all of the execution times are higher and the landscape shows higher variation in execution times at close parameter values. Note that the colour map has the same scale in both figures and that the axes are identical.

The remaining figures show slices of the landscapes in Figures 1 and 2 to paint a more detailed picture of the results. Since Figures 1 and 2 do not give us any reason to suspect a significant difference in the shape of a slice depending on its location in the landscape, we arbitrarily choose the middle value on each axis to set as a constant for each slice.

The Figures 3, 4, 5, and 6 show that the execution time for training both the LDA model and the LR model can be fit linearly very closely against both n and p, with R^2 values exclusively greater than 0.94. Figure 5 does not confirm the quadratic dependence of the training time on the number of predictor variables given by Equation 9, as the quadratic fit and the linear fit are approximately equal across the complete range of the number of predictor variables.

The more variable nature of the training time for LR than LDA for varying n and p is also confirmed by the figures, as the LR curves are more jagged than the curves of LDA.

In Figures 3 and 4, there is a drop in training time from 110,000

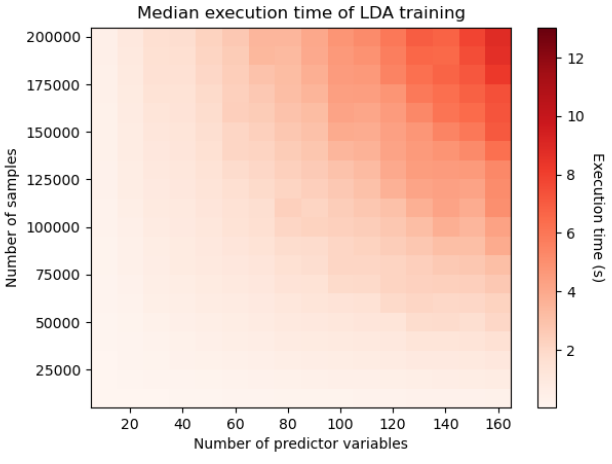


Fig. 1: Median execution times of training the LDA model with varying sample sizes and numbers of predictor variables.

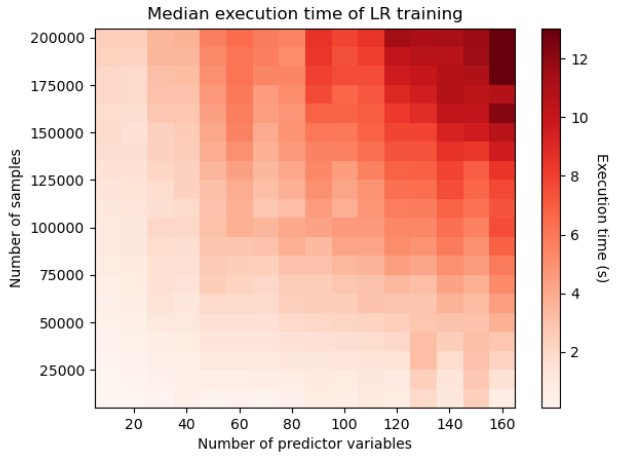


Fig. 2: Median execution times of training the LR model with varying sample sizes and numbers of predictor variables.

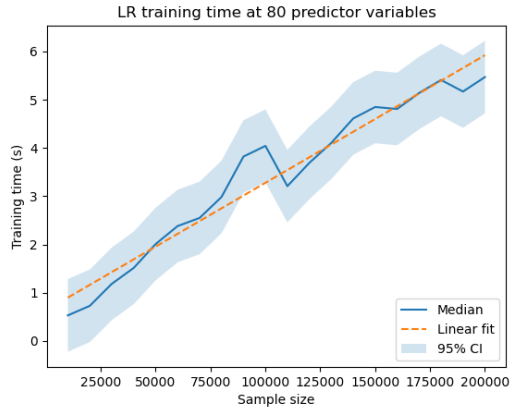


Fig. 4: Execution time of training the LR model with 80 predictor variables. The linear regression line describes the equation $y = 2.644 \cdot 10^{-5}x + 0.6353$ and has $R^2 = 0.9499$.

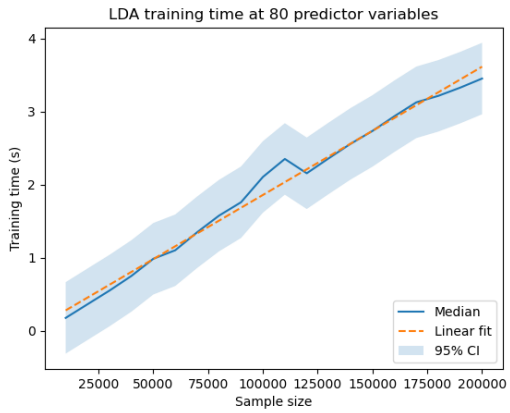


Fig. 3: Execution time of training the LDA model with 80 predictor variables. The linear regression line describes the equation $y = 1.754 \cdot 10^{-5}x + 0.1058$ and has $R^2 = 0.9879$.

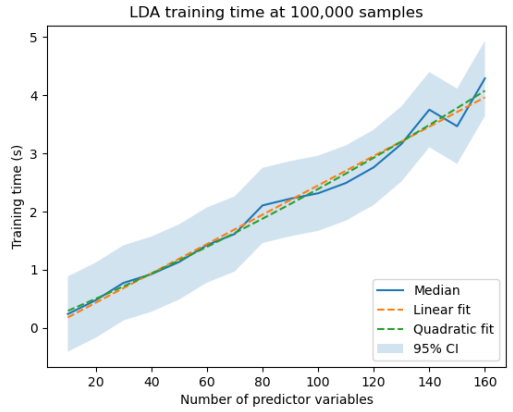


Fig. 5: Execution time of training the LDA model with 100,000 samples. The linear regression line describes the equation $y = 0.02523x - 0.07262$ and has $R^2 = 0.9879$. The quadratic regression line describes the equation $y = 3.262 \cdot 10^{-5}x^2 + 0.01969x + 0.09376$ and has $R^2 = 0.9845$

.

to 120,000 samples. A similar, though less prominent drop can be observed in Figures 5 and 6 from 140 to 150 predictor variables.

5 Discussion

Here, we present a discussion of the reviewed papers and the time complexity analysis, giving rise to a guideline on when to use which model, given in section 6.

5.1 Comparison Papers

Lei and Koehly [11], Liong and Foo [12], and Pohar, Blas, and Turk [15] all came to slightly different conclusions on the performance of LDA vs. LR based on their respective studies.

Pohar, Blas, and Turk [15] and Liong and Foo [12] disagree on the relative performance of LDA when the normality assumption is satisfied. The former found that in this case, LDA performs slightly better than LR. Only when the sample size grows large does LR perform equally well. Liong and Foo, however, found that even when the independent variables are normally distributed LR outperforms LDA. To come to this conclusion they used a dataset of only 25 observations and an unknown sample size, whereas Pohar, Blas, and Turk use sample sizes ranging from 40 to 1000. According to Pohar, Blas, and Turk [15], LDA performs better than LR when using small sample sizes which was not the case for Liong and Foo [12].

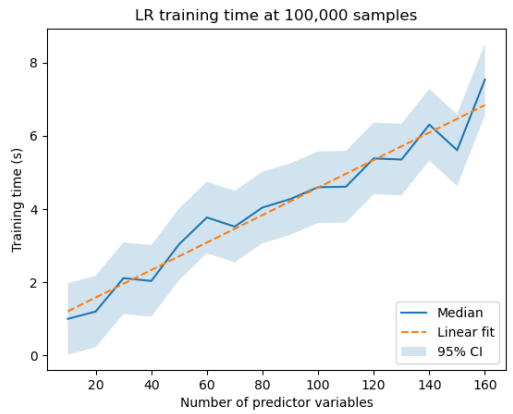


Fig. 6: Execution time of training the LR model with 100,000 samples. The linear regression line describes the equation $y = 0.03755x + 0.8352$ and has $R^2 = 0.9514$.

Liong and Foo [12] also found that using computed prior probabilities according to group sizes as opposed to equal ones increased the performance of LDA in terms of B-index and percentage of correct classification. This coincides with Lei and Koehly [11] who recommend LDA with proportionate prior probabilities and a cut-score of 0.5 for the least amount of wrong classifications.

5.2 Empirical Time Complexity Analysis

The observed drops in training time in Figures 3 to 6 are likely the result of a variable number of background processes occurring concurrently with the training of the models. This variation may have occurred due to a separation of the computations into batches, executed at different times. The models were trained sequentially with the same data sets in an alternating fashion, explaining why the drop is in the same location both in the LR and in the LDA case.

Figures 1 and 2 show that training an LR model takes longer than training an LDA model for all the tested parameter settings. Furthermore, the linear regression lines in Figures 3 to 6 show that training an LR model is likely to remain slower when the parameters grow. When growing the number of predictor variables, LR is projected to be about 1.49 times slower than LDA. When growing the sample size, LR is predicted to remain about 1.51 times slower than LDA. Due to these results being approximately equal, we can say with some certainty that a speed difference of 50% is to be expected across the board. That being said, if we assume that the quadratic regression curve in Figure 5 is more representative than the linear fit, the training time of LR is expected to become shorter than that of LDA for all $p \geq 587$.

Limitations Unfortunately, whether the quadratic fit or the linear fit best describes the observations in Figure 5 cannot be extracted from this work. Due to the large number of experiments that were carried out, we have not been able to capture the behaviour of the LDA model training time at numbers of predictor variables large enough to draw any strong conclusions in this regard. Nonetheless, we can conclude that LDA training is likely to be faster than LR training for use cases with numbers of predictor variables below 587, which is substantial.

A great limitation of this work is the large number of fixed parameter settings, i.e., assumptions, that were made for the models. For the LDA model, 7 parameters were fixed across the experiments. For the LR model, this number was 14. Although these values were chosen either to be optimized for large data sets or to represent most use cases, they may be completely inapplicable in certain scenarios. Furthermore, fixing the number of outcome variables at 2 allowed us to enrich the results of the reference papers [11, 15, 12], but this too is inapplicable in many use cases.

Finally, while the use of synthetic data sets allowed for great comparability of the results, it remains uncertain whether these results would be representative when a real data set is used.

6 Conclusion

With this work, we made an attempt to provide guidelines indicating whether to use LDA or LR given varying use case characteristics, with regard to the training time of each of these methods. In this section, we present these guidelines, based on the combined findings of this work and [11, 12, 15].

The choice between LDA and LR depends on various factors. Following the three papers of Lei and Koehly [11], Liong and Foo [12], and Pohar, Blas, and Turk [15], no clear conclusion on the importance of following the normality assumption can be reached. However, since LDA has been specifically created with this assumption in mind, it may still be advisable to follow it.

In general, LDA performs better than LR when the sample size is small, so in this case the use of LDA is advised. When the goal is to decrease the classification error for large groups, the recommended methods are LR or LDA with extreme priors and a cut-off score of 0.5. When the goal is to decrease the classification error for small groups, the recommended method is LDA with equal priors and a cut score of 0.1. When the goal is to decrease the overall classification error, LR or LDA with proportional prior probabilities and a cut-off score of 0.5 is recommended. It is good to keep in mind that using LDA with proportional prior probabilities only makes sense if the sample is large enough and therefore representative of the population. In the cases where the performance of LDA and LR does not differ much, the use of an LDA model is advisable, due to the projected 33% saving in computation time.

7 Future Work

Throughout the writing of this work, several points have emerged that call for a deeper dive into the matter. Most notably:

- investigating whether the results hold up in more specific use cases, i.e., with different parameter settings of the models;
- investigating whether LDA training times follow a linear or a quadratic curve for an increasing number of predictor variables;
- expanding the research to data sets and models with more output variables;
- investigating whether the results with simulated data hold up when real data sets are used;
- comparing the effectiveness of LDA and LR in broader use cases, for example, as part of a neural network;
- assessing the false positive rate and the true positive rate of LDA and LR with varying biases by inspecting their receiver operating characteristics (ROC) curves.

Acknowledgements

We wish to thank expert reviewer M. Biehl for the idea for the paper and for his valuable feedback. Additionally, we would like to thank B. Popescu and S. Brouwer for their helpful reviews.

References

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] T. Amemiya. *Advanced econometrics*. Harvard university press, 1985.

[3] D. Cai, X. He, and J. Han. Training linear discriminant analysis in linear time. In *2008 IEEE 24th International Conference on Data Engineering*, pages 209–217, 2008.

[4] J. S. Cramer. The origins of logistic regression. 2002.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[6] R. J. Freund, W. J. Wilson, and P. Sa. *Regression analysis*. Elsevier, 2006.

[7] F. E. Harrell and K. L. Lee. A comparison of the discrimination of discriminant analysis and logistic regression under multivariate normality. *Biostatistics: Statistics in Biomedical, Public Health and Environmental Sciences', North-Holland, New York, United States*, pages 333–343, 1985.

[8] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[9] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[10] A. J. Izenman. *Modern multivariate statistical techniques*, volume 1. Springer, 2008.

[11] P.-W. Lei and L. M. Koehly. Linear discriminant analysis versus logistic regression: A comparison of classification errors in the two-group case. *The Journal of Experimental Education*, 72(1):25–49, 2003.

[12] C.-Y. Liong and S.-F. Foo. Comparison of linear discriminant analysis and logistic regression for data classification. *AIP conference proceedings*, 1522(1):1159–1165, 2013.

[13] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[14] T. G. Nick and K. M. Campbell. Logistic regression. *Topics in biostatistics*, pages 273–301, 2007.

[15] M. Perme, M. Blas, and S. Turk. Comparison of logistic regression and linear discriminant analysis: a simulation study. *Metodoloski zvezki*, 1(1):143, 2004.

[16] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[17] H. Scheffe. *The analysis of variance*, volume 72. John Wiley & Sons, 1999.

Opportunities and Challenges in the Adoption of Function-as-a-Service Serverless Computing

Bjorn Pijnacker Jesper van der Zwaag

Abstract— Function-as-a-Service computing is a paradigm in the recent serverless computing architecture that allows a developer to deploy functions without needing to be concerned with underlying deployment specifics. Being quite a recent paradigm, many challenges and opportunities have been discussed in research and industry, and many advancements have been made. In our paper, we give an overview of the change in the state of the art since 2017. To achieve this, we performed a literature search based on two influential papers in Function-as-a-Service research. Literature is investigated for the challenges and opportunities they expose. We outline our results in chronological order, as to analyze how the challenges and opportunities have changed in recent years. Our results indicate that many of the challenges that were present in the beginning of Function-as-a-Service still exist.; however, there is a clear progression in the field to mitigate the issues. As the Function-as-a-Service paradigm has matured, more developers have gotten involved and new difficulties emerged. We conclude that the Function-as-a-Service ecosystem has matured over the years, and a lot of effort has been put into solving some of its problems, but a considerable number of issues still remain.

Index Terms—Function-as-a-Service, serverless computing, FaaS, cloud, edge computing.

1 Introduction

In the age of as-a-Service, much of the burden of managing application and their environments has been lifted from developers, with services instead being managed by the cloud provider. Serverless computing is a newer paradigm that aims to shift this load away from developers completely, instead offering Backend-as-a-Service (BaaS) and Function-as-a-Service (Faas) [1]. FaaS has had a lot of attention in recent years, and there has seen a steady rise in use since the first large cloud vendor, Amazon Web Services, launched a FaaS offering in late 2014 [2]. Since then, FaaS has had its victories and challenges in use and implementation. Previous work explains issues and common pitfalls [3], or investigates the current use-cases of FaaS [4], [5].

In our paper we will give an overview of the current state of the art of Function-as-a-Service, combining the issues named in research with the shown-successful use cases that are demonstrated. Serverless Computing and FaaS are quite recent innovations, and some earlier papers may name problems that have already been solved, or on which the perspective in literature has changed. Using the analysis of these papers we construct an overview of how the story has changed over the years. Concluding, we will give a glimpse into a possible future of Functions-as-a-Service. Has the field progressed enough in the past years to be promising, or is much work required for FaaS to succeed as a paradigm?

In order to perform a comparison between the older state of the art and newer papers we must decide how to construct the older state of the art. For this goal we consider two quite influential papers: "A mixed-method empirical study of Function-as-a-Service software development in industrial practice" [4] and "Serverless Applications: Why, When, and How?" [5]. These papers were among the first to map the entire state-of-the-art for serverless computing and FaaS, both in industry and academia, using a mixed-method approach and by reviewing existing serverless software products. As such, we will be using them to map our 2017 state of the art.

We will answer the following concrete question in this research:

> What challenges and opportunities in Function-as-a-Service computing have been identified, and how have these changed since 2017?

- *Bjorn Pijnacker and Jesper van der Zwaag are with the University of Groningen.*
- *Email: { b.pijnacker.1, j.r.van.der.zwaag }@student.rug.nl.*

We expect many older problems of FaaS to have been solved in the past years, and the ecosystem regarding it has—at least somewhat—matured. We also expect new problems to have emerged as FaaS has gained more widespread usage, and we expect the set of use-cases in which FaaS excels to have become more clear over the past years.

The rest of this paper is structured as follows. In Section 2 we provide a little more background on serverless computing and function-as-a-service. In Section 3 we outline the methods that we will use to do our research, whose results we show in Section 4. Consequently, we discuss our results in Section 5. In Section 6 we review any threats to validity, after which we conclude our research in Section 7. We finally reflect on possible future work in Section 8.

2 Background Information

Offerings in the serverless computing architecture typically fall into either of two groups: Backend-as-a-Service (BaaS) and Function-as-a-Service (FaaS) [1]. BaaS allows developers to outsource their entire backend to a service. This service then takes care of activities such as database management, cloud storage, user authentication, push notifications [6].

Function-as-a-Service (FaaS), on the other hand, is a cloud computing service that allows the user to run a function without manually managing computational resources. It has some great selling points: First, it works on-demand, i.e. code is automatically executed when it is triggered by an external (often HTTP) event. Secondly, as the user does not manage any resources, it can scale automatically based on demand. This makes it very suitable for bursty and irregular workloads. The cost also scales automatically, as it uses the pay-as-you-go model, which costs are only incurred for the computing power and time used, and not for some compute node that may not have been used, as may be the case with Kubernetes- or Platform-as-a-Service (KaaS/PaaS), Infrastructure-as-a-Service (IaaS) and other such more traditional paradigms [7].

FaaS and BaaS are instances of "serverless computing", which means that the cloud provider manages and allocates computational resources for its customers. It does not mean that no servers are used, as they are still needed to execute the code; instead, it means that the customer is not responsible for or concerned with servers and their management. FaaS is one of the most common serverless computing offerings, with its alternative (BaaS) being used by 12% of serverless applications in a recent survey of serverless applications [5].

3 METHODS

To answer our research question as set out in the introduction, we will be performing a short literature review. To find a set of papers to review, we use a forward snowballing approach based on a seed-set of papers. This seed-set consists of two papers: "Serverless Applications: Why, When, and How?" [5] and "A mixed-method empirical study of Function-as-a-Service software development in industrial practice" [4]. The importance of these papers was discussed in the introduction.

Snowballing and filtering consist of multiple steps. After performing the snowballing, we filter based on title and abstract to remove papers that are not relevant. Afterward, we read each remaining paper and remove papers that do not bring a clear challenge or opportunity to light. In our case, we find many secondary papers and relatively few primary ones.

From there we will start by giving an overview of the state of the art according to our seed papers ([4], [5]). This aims to identify a base set of challenges and opportunities. We then continue by viewing newer papers to investigate which problems have been solved, which new problems have emerged, and what—in general—changed. We will extract identified challenges and advantages in each paper, and order these by date. Using this set of challenges and advantages, we analyze whether there is a trend that emerges, whether new issues are introduced and whether issues may disappear from the discussion.

4 RESULTS

In this section, we outline the results found by investigating the selected papers. In Section 4.1 we outline the challenges and advantages in our two seed papers to construct a 2017 state of the art. Proceeding, we look (chronologically) at the papers selected by snowballing, and outline the challenges and advantages that each paper discusses.

4.1 2017 State of the Art

Our first paper in consideration has done a mixed-method empirical study of FaaS software development in industry practice [4]. Of particular interest to us is their third research question: "What are the major advantages and challenges of using serverless and FaaS in practice?"

Leitner et al. [4] state that there are three main classes of advantages: business/cost-, technical-, and security-related advantages. In the business/cost class, a couple of key advantages are identified: (1) the pay-as-you-go pricing model guarantees costs correspond to usage; (2) FaaS liberates developers from managing servers, meaning they can spend more time on business features; (3) per-request billing enables simple cost optimization: faster execution corresponds exactly to costs saved. Some technical- and security-related advantages are also identified: (1) elastic scalability of functions; (2) failover capabilities; (3) increased security, as server management responsibility is shifted to the cloud provider; (4) DDoS is now a billing rather than an availability issue.

Some key challenges are also identified. Leitner et al. [4] name that these might be attested either to the relative immaturity of FaaS while others may be the result of concepts and practices underlying FaaS. The main identified challenges are: (1) lack of tooling (e.g. testing, deployment); (2) difficulty in integration testing; (3) vendor lock-in; (4) container start-up latency; (5) managing function state and tail latency. Note that these are named in decreasing order of frequency of identification by their interviewees.

Eismann et al. [5] investigate why, when, and how to use FaaS by investigating 89 descriptions of existing cases. They report that the most common reasons to implement FaaS are to save costs for irregular workloads and to avoid operational concerns, something which [4] also identified.

No key advantages or challenges are distinctly identified, although Eismann et al. do name that serverless applications are most often used for short-running tasks with low data volume and bursty behavior; however, it is shown that examples of the use of serverless computing exist across all types of applications, such as latency-critical workloads or high-volume core functionality.

4.2 Post-2017

Using forward snowballing we have identified papers that have been used for our analysis. We now proceed paper-by-paper, old to new, and summarize key challenges, opportunities, or other information for each paper.

In [8], authors aim to answer the research question "What are the challenges and drivers motivating researchers to engineer new or extend existing FaaS platforms and platform-specific tools?" In their work, they look at many research papers and analyze which challenges are targeted by these papers. They find the main challenges which lead current research to be (1) function execution performance; (2) function execution security; (3) development environment support; (4) testing and observability; (5) benchmarking; (6) cost optimization; (7) programming models.

The authors of [9] have explored the (technical) factors that influence the decision of using FaaS or not, based on previous work and their own experience. The main takeaways are that a function itself should be stateless, as there is no control over where it's executed. It should also be idempotent: in case of an error, a function is re-executed, and that should not lead to unexpected behavior. The architecture should be event-driven, as that is where the FaaS paradigm is best suited for. Other architectures are likely to not work. Functions need a limited execution time as well as memory consumption and artifact size, the size of the deployed function. Although many environments and languages are supported, FaaS does not support all execution environments natively. As a function needs a brief but noticeable moment to start, called a cold start, low latency cannot be guaranteed. Updating a function is quite trivial, as all maintenance and orchestration are externalized. Different vendors have different ways of interfacing with and deploying a function, sometimes paired with additional platform-specific services. Transferring from one to another vendor might therefore not be effortless. Lastly, FaaS is not best for every workload. Due to FaaS' theoretical unlimited scalability, it is very well suited for unpredictable and bursty workloads. With deploying VMs, one would quickly run into under- or overprovisioning. With regular workloads, FaaS could be used but provides no advantage over conventional deployment models; due to FaaS' cost model, in those cases, FaaS may not be the most cost-effective.

In [10], best/bad practices, and open issues with FaaS are discussed. Asynchronous calls to and between functions can increase the complexity of the system, as do functions calling other functions as well. Avoiding asynchronous calls and reducing functions calling functions, e.g. by merging functions when possible, can reduce complexity. Shared code as well as the usage of too many libraries, can incur problems with the image size as well as long start times. Writing independent, decoupled functions with only needed libraries can solve this. Too many technologies and functions can also increase complexity and maintenance difficulty. Reusing what exists can help with that. Some of the open issues are that developers need to understand the event-driver paradigm that is best suited for FaaS. Tools to develop and deploy functions easily are also not matured yet. Testing is also a problem. Unit testing a function itself is easily done, but testing how the function would behave in the complete system, is more difficult.

In [11], Prechtl, Lichtenthäler, and Wirtz investigate the security of different installable FaaS platforms[1]. The security of these cloud installations is investigated by examining the security features that these platforms provide. This examination is based on a threat model and trust assumptions made in their paper. The platforms that are investigated are Kubeless 1.0.5, OpenFaaS 0.18.2, Fission 1.7.1, Knative 0.12.1, and OpenWhisk 0.9.0. They find the main challenges are that there has been little research into security for FaaS, that many cloud systems are not very mature with regards to security, and that many advanced security features are often not present in the discussed installable cloud platforms.

The paper in [12] concerns itself with patterns for Function-as-a-

[1] An installable cloud platform is one that can be hosted on private infrastructure. This is in contrast to hosted cloud platforms, where the user is not concerned with hosting the actual cloud software [11].

Service computing. The paper gives a summary of programming patterns and when to use them based on a multivocal literature review. They find 32 patterns that fit into five categories: orchestration and aggregation, event management, availability, communication, and authorization. We can conclude some general challenges noted in their work: there is a lack of patterns in general, and patterns are not always very clear; different patterns are required per vendor. For example, AWS Lambda provides a queuing service (SQS) that allows for using FIFO messaging, while in Azure, FIFO messages are something the developer must manage themselves. We find also that some patterns exist in order to manage other issues. As an example: the function warmer pattern aims to "solve" cold start delay, by ensuring the function never goes cold in the first place. This, of course, conflicts with some of the advantages, such as paying only for runtime, as the function warmer pattern invokes the function more than necessary.

In [13], the authors summarize recent advancements in the mitigation of cold start delay for serverless computing. While no specific challenges or advantages can be concluded from their work, their summary of recent advancements does provide good insight. We can see that much research into cold start has been performed and that there has likely been much evolution related to cold start mitigation.

In [14], the opportunities and challenges of serverless edge computing are discussed. The pay-per-use model is excellent for any non-regular workload. With this model, servers do not need to be on all the time which can save energy consumption. It works great for event-driver applications, as FaaS is well suited for that paradigm without introducing much complexity. Due to its statelessness, it is also very portable and can be executed theoretically anywhere with supported hardware. This separation of state and computation also allows for excellent separation of concern and individual scalability. FaaS is also very easily parallelizable, as it can be called many times ad-hoc, without the concern of orchestration and managing servers. One of the most common issues is the cold starts. Although, some promising solutions are proposed to at least reduce the problem. It also is not well suited for continuous workloads compared to more conventional deployment models, especially concerning price and statelessness. FaaS is also fitted for short and CPU-bound workloads, so longer and IO-bound workloads, e.g. edge AI, may not be cost-efficient with FaaS. Also, not knowing exactly where which function will be executed, can make resource optimization more complex and increase unnecessary network communication. Security is also not fully in the hand of the developer, as the VMs and servers are not in their control. Additionally, most serverless offerings do not support specialized hardware, i.e. GPUs and FPGAs. IoT's most common communication pattern, MQTT, cannot trigger functions, which use standard HTTP triggers. Simulating how functions would behave in the production environment is also very difficult, which can hinder debugging and testing. Lastly, differences between vendors' use of FaaS can also make it complicated to switch between vendors.

The survey conducted in [15] has examined 275 papers. They found that there is significant growth in serverless computing as a research topic, but there are both benefits and drawbacks. The most common benefits include the cost model (pay-per-use), autoscaling, no server-side management, easy deployment, and potential latency decrease, as functions can be executed on a server near where it is requested. The latency can also be an issue, due to cold starts. The cost model can also be a drawback for IO-bound functions. Functions that require specialized hardware, lots of computational power, or a long running time, are also not likely to work on FaaS platforms. Debugging and testing are also more difficult, as well as switching between vendors. Migrating from legacy systems to serverless can also be difficult as not much research has gone into this. Functions are also not predictable in startup time or performance. Due to its statelessness, it is also hard to track interaction between all serverless components. Heterogeneous hardware, e.g. GPUs, is also barely supported. Future research can be done in the cold starts, as that is a problem for many users. Also, the QoS, pricing models, and migration from legacy systems are not researched enough yet. Lastly, developers need better debugging, testing, and benchmarking tools.

In [16], several pros and cons are discussed. One of the pros in NoOps, no operations. There is just code that is deployed, without server management or orchestration. This also allows for fast and independent deployment, also because functions are deployed independently. The pay-per-use model can also be a pro, as costs for running the function are equal to the time and resources used, without any infrastructural costs. Moreso, there are no resources in use when a function is not running, meaning costs are 1:1 with function runtime. Functions also scale automatically. One of the main cons is testing and debugging. This is harder and more complex compared to legacy systems. Long-running processes are also not supported. Performance is unpredictable and there can be a startup latency, known as the cold start. Switching between vendors can also be difficult. And FaaS is tailored toward the event-driven paradigm, which may be new and complex to developers. Due to its short life, not many ideal or optimal architectural patterns have been established.

Targeting industry more than academia, [17] looks at common topics in StackOverflow questions in serverless computing. According to the authors, "[StackOverflow] is a popular Q&A forum for developers to seek advice from peers when they have programming issues." They find many topics on which questions are asked, though the main ones are package integration, function invocation, performance, conceptual difficulty, version compatibility, and configuration.

In [18], Wen et al. compare four commonly used FaaS offerings in terms of startup latency, performance, and concurrency performance. While the exact results of their comparison is not of great importance to our research question, we can conclude some global challenges from their paper. For instance, they find that programming languages are limitedly supported, that the programming language choice can have wildly different cold start times and that package size limits are also in place. Other challenges they name are that each platform has different limitations on compute and memory limitations and that there is no support for GPU-based libraries.

The paper in [19] investigates the effect of different deployment strategies for reducing the cold start delay of AWS Lambda. The main challenge we can conclude from this paper comes from the many choices for language runtime, memory configuration, and deployment type. The combination of these has an effect on the cold start delay, which may not always be clear to the developer of the software. Moreso, since there is active research into reducing cold start delay, we can conclude this to be an active issue in 2022. In their paper, they give a few recommendations to developers for reducing the cold start delay.

In [20], FaaS is explored in the open-source community. Despite the growing popularity, not many mature projects use the model, mostly due to difficult migration. The independents of functions, which should allow for flexible software development, is not used much in practice. The use cases for FaaS are very limited due to (a combination of) the limitations concerning the package size, amount of computation power and memory, and its short-lived lifespan. Often external resources and plugins are used. FaaS has received a lot of attention through the years, but it is still struggling with a number of issues.

5 Discussion

In Section 4.2 we see many challenges come to light over the years. While we see some challenges pop up here and there, it is very evident that most of the research concerns cold start delay, and mitigation strategies for it, with nine out of thirteen papers concerning themselves with it. This is one of FaaS' major issues, but fortunately, there are mitigation strategies and researchers are also working on the subject [13], [19]. AWS Lambda, the most used FaaS offering, also has some mitigation strategies, e.g. Lambda SnapStart [21] to improve Java startup time by up to 10x, and reusing executing environments to improve performance, or the ability to keep connections alive [22]. A related challenge is package or artifact size. Three of the four papers mentioning the limited size find this to be a factor in cold start delay, with a larger package size influencing the latency for containers to start. Moreso, some cloud providers have limitations on the size of an artifact. If the artifact is too big, the provider will refuse to accept it.

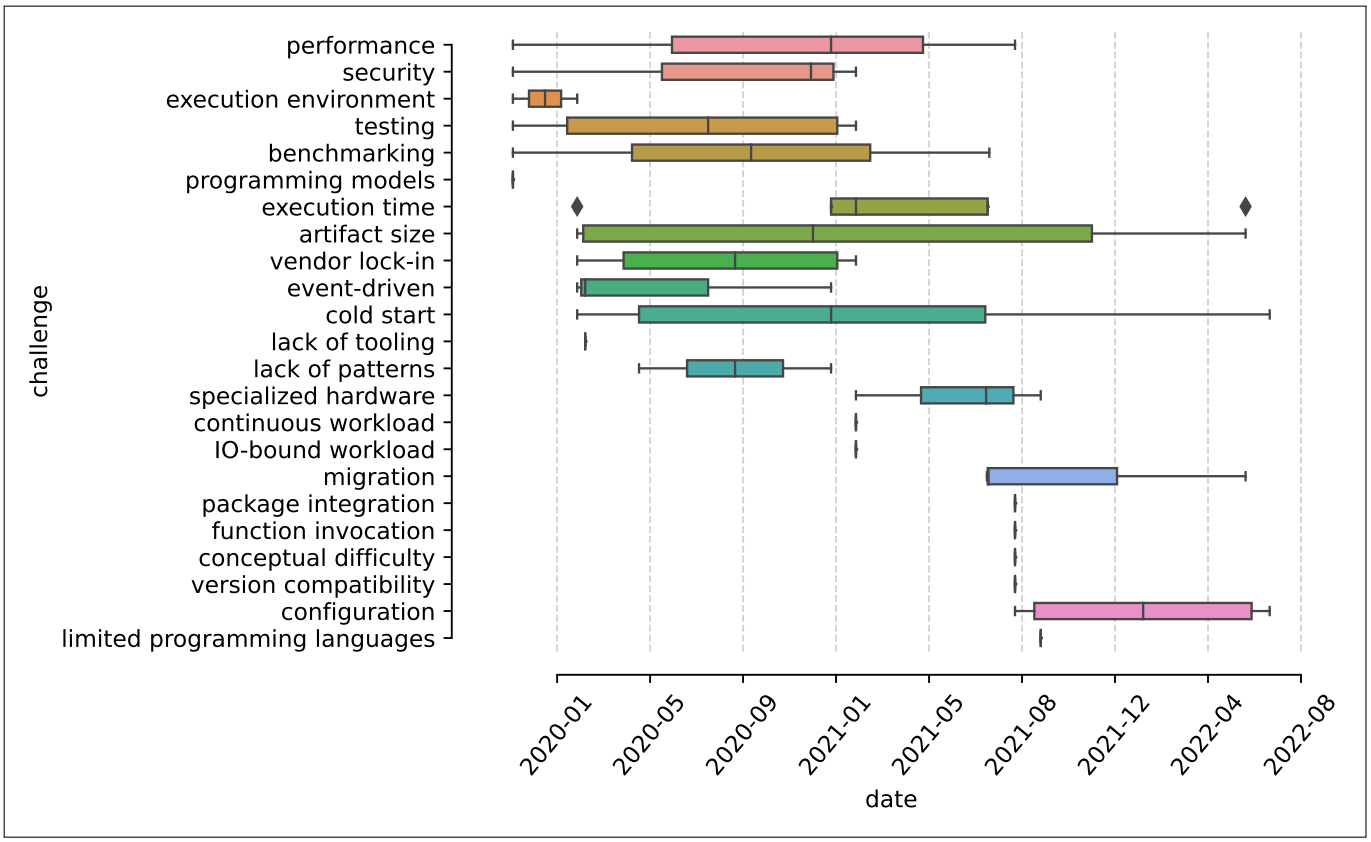


Fig. 1. Identified challenges in Section 4.2 against the publishing date of the paper(s) the challenge appears in

However, there are some solutions and workarounds to this problem, e.g. storing needed data/files externally and reducing the package size to only necessities [22], [23].

In addition, execution time is also a limitation for users, it is mentioned 5 times. The execution time limit used to be 5 minutes for AWS, but it has been increased to 15 minutes since October 10th 2018 [24]. For Azure, this depends on the user's plan: this can range from 10 minutes to unlimited runtime. However, if the function is triggered using the HTTP trigger, it must respond within 230 seconds [25].

Being mentioned four times in our set of papers is vendor lock-in. This is mentioned sometimes specifically as an issue, though other times in the context of a solution or another piece of literature. For example, some patterns are required specifically for tools belonging to a single vendor [12]. Interesting to note is that there are already companies working on tooling that can run serverless code on multiple cloud platforms without needing to change the code [26].

Another challenge we see multiple times concerns the security of FaaS, with three out of thirteen papers naming it. The last challenge that is named multiple times is that of (irregular) function performance. Other identified issues include difficulty in tooling and testing and several configuration challenges, though these were less often discussed that previously mentioned challenges.

One of the main advantages of FaaS mentioned in the papers is the cost model: a users only pays for what is used. This works great in combination with one of FaaS' main features: automatic scaling. It works great for any non-regular workload without worrying about under or over-provisioning VMs. It takes away any server-side management and orchestration, which allows for easy, NoOps, deployment. Lastly, it could potentially reduce latency by automatically running the function as close to a user as possible, since large cloud providers may have servers distributed all over the world. In our opinion, a very interesting result is that the advantages discussed in the papers have not changed. The papers concern themselves mainly with targeting issues that make serverless computing more difficult to adopt, while advantages have remained static over time.

In Figure 1 we show all identified challenges against the papers' publishing dates. This shows us that many of the challenges are discussed quite regularly in literature, with only some having a significantly different time span. For example, challenges regarding migration and configuration are discussed only in newer literature. This can likely be explained due to the recent growth in FaaS. As more inexperienced developers start experimenting with it and integrating it into projects, more questions regarding migrating old projects or the configuration of FaaS environments will crop up.

6 Threats to Validity

The mean threat to the validity of our study concerns the sample size and the acquisition method of the papers in our literature study. In our results we see many issues crop up which are named only once or a handful of times. A larger set of papers would produce more accurate results. Moreso, our papers were acquired by snowballing from two seed papers. Due to this, follow-up research that cites these papers may be skewed toward the subjects in the seed papers. By doing a more widespread query search, results would be more representative of the complete serverless and FaaS ecosystem.

7 Conclusion

In our paper, we investigated how the challenges and opportunities in the Function-as-a-Service paradigm have changed in the past few years. For this purpose, we performed snowballing on two influential papers and investigated challenges and opportunities named in each collected paper.

To reiterate, our research question states: "What challenges in Function-as-a-Service computing have been identified, and how has this changed since 2017?".

We found that there were no notable changes in opportunities or advantages in FaaS over the last few years. FaaS has some intrinsic advantages that were identified from its start, and no fundamental changes have been made to the FaaS paradigm itself since then. The only change we see are improvements in mitigating the issues and problems surrounding its adoption.

We found the main challenges to be with the cold start latency of functions and with the execution time of functions. Other noteworthy issues are testing support, FaaS security, artifact size, and vendor lock-in.

In our introduction we stated the following hypothesis:

> *"We expect many older problems of FaaS to have been solved in the past years, and the ecosystem regarding it has—at least somewhat—matured. We also expect new problems to have emerged as FaaS has gained more widespread usage, and we expect the set of use-cases in which FaaS excels to have become more clear over the past years"*

We see that this is–for the most part—true. From our results and discussion, we can conclude that the focus has not changed much in the past few years. We see that the main issues cropping up in our analysis of the 2017 state of the art are also discussed in later papers. However, we do see a clear progression in the field. Many of the challenges named are also noted by cloud providers in the industry. For example, AWS has made an effort of mitigating many of the current issues. While no issues have necessarily been solved, we do see some newer issues crop up in recent years that are indicative of inexperienced developers joining the FaaS ecosystem; namely questions regarding the migration of non-FaaS code and configuration of FaaS systems.

8 Future Work

While the main issues noted concern cold start latency and function execution time, these are already being targeted by academic research and by industry cloud providers. One aspect that we think can use more work is the tooling that is provided for FaaS, both in development, testing, and debugging. Another area where future work can be completed is that of security. Security is currently unclear and not always well-implemented. Both in FaaS offerings and in installable cloud platforms this could definitely be improved.

Acknowledgements

We would like to thank our expert reviewer Prof. Dr. Vasilios Andrikopoulos and our peer reviewers Carlos Brito Gonzalez and Hessel van Oordt for their insightful comments which helped tremendously in improving our paper.

References

[1] Red Hat. "What is serverless?" (May 11, 2022), [Online]. Available: https://www.redhat.com/en/topics/cloud-native-apps/what-is-serverless (visited on 02/19/2023).

[2] Amazon Web Services, Inc. "Amazon Web Services Announces AWS Lambda," Press Center. (Nov. 13, 2014), [Online]. Available: https://press.aboutamazon.com/2014/11/amazon-web-services-announces-aws-lambda (visited on 02/22/2023).

[3] J. M. Hellerstein, J. Faleiro, J. E. Gonzalez, *et al.* "Serverless Computing: One Step Forward, Two Steps Back." arXiv: arXiv:1812.03651. (Dec. 10, 2018), [Online]. Available: http://arxiv.org/abs/1812.03651 (visited on 01/17/2023), preprint.

[4] P. Leitner, E. Wittern, J. Spillner, and W. Hummer, "A mixed-method empirical study of Function-as-a-Service software development in industrial practice," *Journal of Systems and Software*, vol. 149, pp. 340–359, Dec. 16, 2018, ISSN: 0164-1212. DOI: 10.1016/j.jss.2018.12.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121218302735 (visited on 01/17/2023).

[5] S. Eismann, J. Scheuner, E. van Eyk, *et al.*, "Serverless Applications: Why, When, and How?" *IEEE Software*, vol. 38, no. 1, pp. 32–39, Dec. 22, 2020, ISSN: 1937-4194. DOI: 10.1109/MS.2020.3023302.

[6] Cloudflare. "What is BaaS? — Backend-as-a-Service vs. serverless," Cloudflare. (), [Online]. Available: https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/ (visited on 03/07/2023).

[7] Red Hat. "What is FaaS?" (2020), [Online]. Available: https://www.redhat.com/en/topics/cloud-native-apps/what-is-faas (visited on 03/07/2023).

[8] V. Yussupov, U. Breitenbücher, F. Leymann, and M. Wurster, "A Systematic Mapping Study on Engineering Function-as-a-Service Platforms and Tools," in *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, ser. UCC'19, New York, NY, USA: Association for Computing Machinery, Dec. 2, 2019, pp. 229–240, ISBN: 978-1-4503-6894-0. DOI: 10.1145/3344341.3368803. [Online]. Available: https://doi.org/10.1145/3344341.3368803 (visited on 02/15/2023).

[9] R. Lichtenthäler, S. Winzinger, J. Manner, and G. Wirtz, "When to use FaaS? - Influencing technical factors for and against using serverless functions," presented at the CEUR Workshop Proceedings, vol. 2575, 2020, pp. 39–47.

[10] J. Nupponen and D. Taibi, "Serverless: What it Is, What to Do and What Not to Do," in *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*, Mar. 2020, pp. 49–50. DOI: 10.1109/ICSA-C50368.2020.00016.

[11] M. Prechtl, R. Lichtenthäler, and G. Wirtz, "Investigating Possibilites for Protecting and Hardening Installable FaaS Platforms," in *Service-Oriented Computing*, S. Dustdar, Ed., ser. Communications in Computer and Information Science, Cham: Springer International Publishing, 2020, pp. 107–126, ISBN: 978-3-030-64846-6. DOI: 10.1007/978-3-030-64846-6_7.

[12] D. Taibi, N. El Ioini, C. Pahl, and J. Niederkofler, "Patterns for serverless functions (Function-as-a-Service): A multivocal literature review," presented at the CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science, 2020, pp. 181–192, ISBN: 978-989-758-424-4.

[13] P. Vahidinia, B. Farahani, and F. S. Aliee, "Cold Start in Serverless Computing: Current Trends and Mitigation Strategies," in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, Aug. 2020, pp. 1–7. DOI: 10.1109/COINS49042.2020.9191377.

[14] M. S. Aslanpour, A. N. Toosi, C. Cicconetti, *et al.*, "Serverless Edge Computing: Vision and Challenges," in *2021 Australasian Computer Science Week Multiconference*, ser. ACSW '21, New York, NY, USA: Association for Computing Machinery, Feb. 1, 2021, pp. 1–10, ISBN: 978-1-4503-8956-3. DOI: 10.1145/3437378.3444367. [Online]. Available: https://doi.org/10.1145/3437378.3444367 (visited on 02/15/2023).

[15] H. B. Hassan, S. A. Barakat, and Q. I. Sarhan, "Survey on serverless computing," *Journal of Cloud Computing*, vol. 10, no. 1, p. 39, Jul. 12, 2021, ISSN: 2192-113X. DOI: 10.1186/s13677-021-00253-7. [Online]. Available: https://doi.org/10.1186/s13677-021-00253-7 (visited on 02/15/2023).

[16] D. Taibi, J. Spillner, and K. Wawruch, "Serverless Computing-Where Are We Now, and Where Are We Heading?" *IEEE Software*, vol. 38, no. 1, pp. 25–31, Jan. 2021, ISSN: 1937-4194. DOI: 10.1109/MS.2020.3028708.

[17] J. Wen, Z. Chen, Y. Liu, *et al.*, "An empirical study on challenges of application development in serverless computing," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2021, New York, NY, USA: Association for Computing Machinery, Aug. 18, 2021, pp. 416–428, ISBN: 978-1-4503-8562-6. DOI: 10.1145/3468264.3468558. [Online]. Available: https://doi.org/10.1145/3468264.3468558 (visited on 02/15/2023).

[18] J. Wen, Y. Liu, Z. Chen, J. Chen, and Y. Ma, "Characterizing commodity serverless computing platforms," *Journal of Software: Evolution and Process*, vol. n/a, no. n/a, e2394, 2021, ISSN: 2047-7481. DOI: 10.1002/smr.2394. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2394 (visited on 02/15/2023).

[19] J. Dantas, H. Khazaei, and M. Litoiu, "Application Deployment Strategies for Reducing the Cold Start Delay of AWS Lambda," in *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, Jul. 2022, pp. 1–10. DOI: 10.1109/CLOUD55607.2022.00016.

[20] N. Eskandani and G. Salvaneschi, "The uphill journey of FaaS in the open-source community," *Journal of Systems and Software*, vol. 198, p. 111 589, Apr. 1, 2023, ISSN: 0164-1212. DOI: 10.1016/j.jss.2022.111589. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121222002655 (visited on 02/15/2023).

[21] "Improving startup performance with Lambda SnapStart - AWS Lambda." (), [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/snapstart.html (visited on 03/12/2023).

[22] "Best practices for working with AWS Lambda functions - AWS Lambda." (), [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/best-practices.html#function-configuration (visited on 03/12/2023).

[23] "Creating and sharing Lambda layers - AWS Lambda." (), [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html (visited on 03/12/2023).

[24] AWS. "AWS Lambda enables functions that can run up to 15 minutes," Amazon Web Services, Inc. (), [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2018/10/aws-lambda-supports-functions-that-can-run-up-to-15-minutes/ (visited on 03/12/2023).

[25] Microsoft. "Azure Functions scale and hosting." (Nov. 22, 2022), [Online]. Available: https://learn.microsoft.com/en-us/azure/azure-functions/functions-scale (visited on 03/12/2023).

[26] Serverless. "Serverless - Infrastructure & Compute Providers." (2022), [Online]. Available: https://serverless.com/framework/docs/providers (visited on 03/12/2023).

Foveated image and video quality metrics: A survey

Sven Veenhuijsen, Sjoerd Hilhorst

Abstract— Foveated rendering is a method to reduce computational cost by rendering pixels more densely in the central (foveal) region of vision and more sparsely in the peripheral region. Image and video quality metrics provide standardized quantitative descriptions of degradation and offer an alternative to costly user studies. Most image quality metrics do not consider the difference in perceived quality between the foveal region and the peripheral region. Foveated image and video quality metrics incorporate this characteristic and are therefore a useful tool in quantifying quality loss in foveated images and videos. This paper reviews the state of the art in foveated image and video quality metrics and provides a guideline for professionals in choosing the most suitable metric for assessing image or video degradation by foveated rendering on a case-by-case basis. Subsequently, a comparison between the methods is made, focusing on their computational efficiency, features and applicability.

Index Terms—image/video quality, foveated rendering, perceptual metric

1 Introduction

With the increasing popularity of virtual reality (VR) and augmented reality (AR), demand for real-time high-quality image rendering increases as well. Since the visual quality of these images is essential, they must adhere to a certain standard. Furthermore, the displays require a high spatial and temporal resolution. This is poses a problem, since hardware gets pushed towards maximum utilization, with rendering times getting increasingly shorter (90 Hz and higher for VR). Significant improvements have been made in display devices, rendering software, and hardware, but computational efficiency and bandwidth are often still significant factors in limiting visual quality. A solution to reduce the computational costs of rendering, while minimizing image quality loss, is called foveated rendering.

Foveated rendering is based on the notion that the human visual system (HVS) is non-uniform. Particularly, in the foveal region, the HVS has high spatial acuity, while in the peripheral region, acuity decreases dramatically. Foveated rendering methods make use of these properties, by degrading images in the peripheral region, to reduce computational costs with no or minimal perceivable loss in image quality by users.

Image and video quality metrics aim to provide a standardized way to quantitatively describe image degradation. These metrics offer an alternative to costly user studies, and could prove useful in the development of cost functions as well, which are essential in machine learning. Unfortunately, most image quality metrics do not consider the difference in perceived quality between the foveal region and the peripheral region. As such, these methods are not suitable for assessment of the quality of foveated rendering methods. However, a small subset of image and video quality metrics acknowledges and incorporates this characteristic. These methods are called *foveated image and video quality metrics*.They take into consideration the loss of visual acuity due to increased eccentricity, and are therefore a useful tool in quantifying quality loss in foveated images and videos.

The aim of this paper is to review the state of the art in foveated image and video quality metrics. This paper provides a guideline to professionals in the field of foveated rendering, for choosing the most suitable metric for assessing image or video degradation by foveated rendering, on a case-by-case basis.

This paper is structured as follows: Section 2 addresses key components of the human visual system, important for understanding foveated rendering, and gives a summary of non-foveated image and video quality metrics. Section 3 provides an overview of the state-of-the-art foveated image and video quality metrics. Section 4 presents a comparison of the metrics, which highlights their strengths and weaknesses. Finally, Section 5 offers the conclusions of our research, and Section 6 suggests possible extensions of the research conducted in this paper.

- *Sven Veenhuijsen is with University of Groningen, E-mail: s.m.veenhuijsen@student.rug.nl*
- *Sjoerd Hilhorst is with University of Groningen, E-mail: s.j.hilhorst@student.rug.nl*

2 Background

To gain an in-depth understanding of foveated image and video quality metrics, we first present a summary of non-foveated image and video quality metrics and address the issues in using non-foveated image and video quality metrics for foveated content. Secondly, we present key components of the human visual system important for understanding foveated rendering.

2.1 Image and video quality metrics

Automated image and video quality assessment have been thoroughly researched. Image and video quality metrics allow quantification of image degradation without the need for user studies. In most cases, a good quality metric has to correlate well with the perception of quality by a human observer. For example, denoising methods find a compromise between noise and blur. Here, an image quality metric should indicate such a compromise that is consistent with human judgment [8].

Simple quality metrics such as Mean Square Error (MSE) or Peak Signal-To-Noise Ratio (pSNR) often fail in this aspect, since these metrics do not correspond well with perceived degradation, making them an inaccurate predictor for perceived quality loss [4]. Structural Similarity Index (SSIM) takes another approach, and is based on the perceived change in structural information, rather than absolute errors, and corresponds much better with human perception [20]. Another approach correlating well with human perception is proposed by Soundararajan et al. [15], which is based on entropic differences in the spatial and temporal domains. Machine learning has also been used for generating quality metrics. Zhang et al. [21] proposes a machine learning-based perceptual quality metric which proves to be remarkably robust. A bottom-up approach taken in many metrics involves modeling the HVS based on psychophysical models. Central in these metrics is often the contrast sensitivity function (CSF). Another advantage of such metrics is that these can be *physically calibrated*, accounting for physical properties of the display, such as brightness, size, viewing distance, and viewing angle. Since foveation-based rendering methods depend on degradation with respect to viewing angle as their main characteristic, all foveated image and video quality metrics incorporate a psychophysical approach. The metrics discussed previously pertain to full-reference quality metrics. These metrics compare a test image with a non-degraded reference image. Another class of

quality metrics, no-reference quality metrics, evaluates the quality of an image without a reference image. Notable no-reference metrics are BRISQUE [9] and NIQE [10]. these metrics primarily make use of measurable deviations from statistical regularities observed in natural images [6].

2.1.1 Differences between image and video quality metrics

In principle, image quality metrics can be applied to videos by for example taking the average of applying the metric over each frame. However, lack of interframe information in the calculation of the metric makes them not appropriate for identifying temporal visual artifacts [8]. Video quality analysis requires a different approach from static images, due to the interaction of spatial and temporal vision. For example, a high-frequency noise could be well visible in a single frame but can disappear in high frame rate video. Therefore, video metrics often model both temporal and spatial aspects of human vision to obtain a more accurate quality metric.

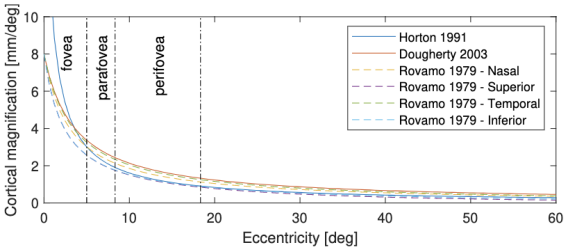


Fig. 1. The magnitude of cortical magnification according to several models [Dougherty et al. [1]; Horton [5]; Rovamo and Virsu [12]. Rovamo et al. model provides different estimates for each part of the visual field [8].

2.2 Human visual system (HVS)

Cortical Magnification

The sensitivity of our vision changes outside the central, foveal region of the visual field, which can be explained by the concept of cortical magnification. This refers to the fact that different portions of the visual field are processed by different numbers of neurons in the visual cortex. The central region is processed by many more neurons per steradian of visual field than the surrounding regions [5]. The cortical magnification factor expresses this relationship in terms of millimeters of cortical surface per degree of visual angle. Several models of cortical magnification proposed in the literature are plotted in Figure 1.

Contrast Perception

Frequently, image and video quality metrics which are based on human perception of visual degradation make use of a contrast sensitivity function (CSF). The CSF is a psychophysical model that describes the sensitivity of the human visual system to different spatial frequencies at a range of luminance contrasts (see Figure 2). It is an important tool used in image and video quality metrics because it helps to quantify the visual quality of images and videos by evaluating the degree to which their contrast and spatial frequency content is perceived by human viewers. By calculating the CSF of an image or video, it is possible to estimate the degree to which the image or video will be perceived by human viewers, and to identify potential issues that may affect the visual quality of the content.

Foveated quality metrics often incorporate the cortical magnification factor in the CSF to mimic the human visual system's natural sensitivity to details in the foveal region. By applying the cortical magnification factor to the CSF, the metric considers the loss of fidelity in peripheral vision. There are numerous CSF models available, the CSF suitable for a particular quality metric depends on the details of the metric. For instance, the metric proposed in [8] is based on peripheral, temporal aspects of human perception, and consequently utilizes a model that incorporates these aspects. A frequently used CSF dependent on spatial frequency and eccentricity is given in [3]

$$CT(f,e) = CT_0 \, exp(\alpha f \frac{e+e_2}{e_2}) \tag{1}$$

where f is the spatial frequency (cycles/degree), e is the retinal eccentricity in degrees, CT_0 is a constant minimal contrast threshold, α is the spatial frequency decay constant, e_2 is the half-resolution eccentricity.

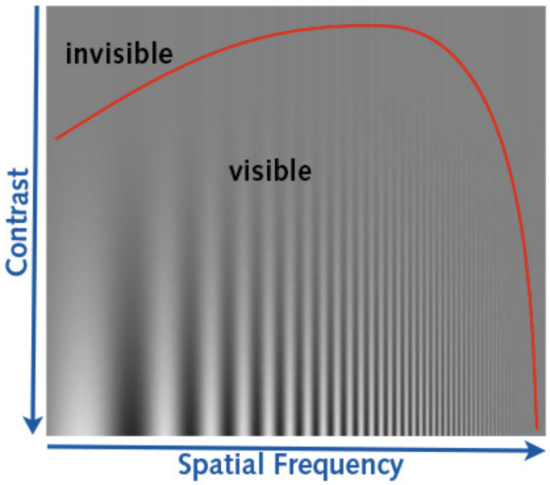


Fig. 2. A typical contrast sensitivity function (red line) as a function of contrast and spatial frequency

Contrast masking

Contrast masking is a phenomenon where the perception of a visual stimulus is affected by nearby visual stimuli [2]. This effect reduces perceived contrast due to the nearby visual stimuli having a similar spatial frequency. An illustration of the effect can be seen in Figure 3, where a Gabor patch is placed on top of a sine grating of similar spatial frequency. Contrast masking is frequently used in quality metrics, as metrics obtain perceived differences between test and reference images using a contrast masking model, as is done in [8]. To account for foveated rendering, metrics can operate on contrasts that are normalized by a detection threshold given by the cortical magnification factor.

3 State of the art

Now that we have established sufficient background knowledge regarding image and video quality metrics and the human visual system, we will discuss different foveated image and video quality metrics.

3.1 Foveated Wavelet Image Quality Index

One of the first foveated quality metrics is FWQI, proposed by Wang et al. [19], operating solely on images. While contrast sensitivity as a function of retinal eccentricity has been used in video/image compression, it is the first method to incorporate it into a quality metric [3]. The method works by defining a function $S(v,f,x)$ depending on the viewing distance, frequency, location, and the cutoff frequency based on Equation 1. This metric is obtained by decomposing the original and test image with the discrete wavelet transform DWT. The function is then multiplied by the difference between the wavelet coefficients for the original and test image. This is done for all wavelet coefficients and then averaged, obtaining FWQI, an index between 0 and 1. The metric is used to develop and optimize EFIC [18], a foveated wavelet image encoding algorithm. The paper presents some images

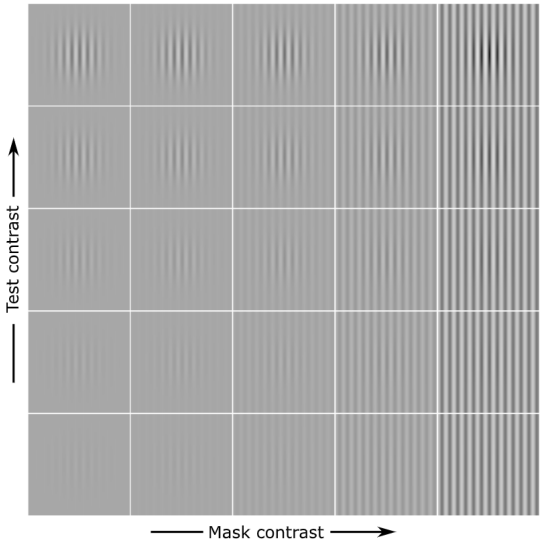


Fig. 3. Illustration of the contrast masking effect. The contrast of a masker (sine grating) is increasing from left to right, making the test contrast (Gabor) more difficult to detect. [8]

obtained using EFIC, and SPHIFT, an ordinary wavelet image encoding algorithm, but no comparison of FWQI to other metrics or user perception has been performed.

3.2 Foveated mean squared error

An improvement on ordinary MSE is proposed by Rimac et al. [11] which takes into account two aspects of the HVS. First is a decrease in contrast sensitivity as a function of retinal eccentricity, this is achieved similarly to FWQI [19]. The second and novel characteristic proposed in the paper is based on spatial acuity dependence on the velocity of an image traveling across the retina.

As mentioned in Section 2, contrast sensitivity is a function of retinal eccentricity. To take into account a reduction in contrast sensitivity away from the fixation point, FMSE introduces a foveation term which is multiplied by the CSF, for a given frequency and eccentricity. This foveation sensitivity function ranges from 0 to 1, where 1 is achieved at eccentricity 0, i.e. at the foveal region. 0 occurs when the given frequency exceeds the maximum highest visible frequency f_{max}, i.e. when the contrast cannot be noticed anymore.

Secondly, the method addresses the influence of movement in foveation-based contrast sensitivity. Concretely, spatial acuity of the visual system depends partly on retinal image velocity $\vec{v_r}$, expressed as follows:

$$\vec{v_r} = \vec{v_i} - \vec{v_t} \tag{2}$$

where $\vec{v_i}$ is the object velocity and $\vec{v_e}$ is the eye velocity. Spatial acuity depends on the retinal image velocity. When retinal velocity increases, the highest visible spatial frequency decreases. FMSE introduces a maximum frequency component f^*_{max}:

$$f^*_{max}(v_r) = f_{max} \cdot \frac{v_c}{v_r + v_c} \tag{3}$$

Where f^*max is the reduced spatial frequency limit, f_{max} is the highest visible spatial frequency for $v_r = 0$, and v_c is the so-called corner velocity, a constant whose value is chosen to fit experimental data. If v_r is above the corner velocity v_c, temporal frequency causes a reduction of spatial resolution.

The foveation-based quality metric FMSE is then obtained by several steps. First, the difference between the reference and test image is taken. Then the difference image is filtered for K frequencies, by convoluting the image by filters obtained by the sensitivity function, for a given frequency f_K.

Video quality metric accuracy was evaluated by correlating the FMSE scores with quality grades given by human viewers, and comparing them with ordinary metrics such as MSE and PSNR. A high correlation was found between FMSE and the subjective grades for various videos, and the metric always scored better than the standard MSE and PSNR metrics.

3.3 Foveal signal-to-noise ratio

Lee et al. [14] convert a video presentation from Cartesian coordinates to a curvilinear coordinate system by a foveation filtering operation and then define and apply a foveated signal-to-noise-ratio to assess quality degradation. An illustration of the mapping between Cartesian coordinates and curvilinear coordinates can be seen in Figure 4.

Let $\phi(x)$ define a mapping of cartesian coordinates $x = (x_1, x_2)$ to curvilinear coordinates $\phi(x) = [\phi_1(x_1, x_2), \phi_2(x_1, x_2)]$ and let S_0 be a region on the original image and A_0 be the area of a region in curvilinear coordinates. Then, $A_0 = \int_{S_0} J_\phi(x)dx$ where $J_\phi(x)$ is the Jacobian of the coordinate transformation $\phi(x)$. With the Jacobian $J_\phi(x)$ we can define the foveated signal-to-noise ratio as

$$FNSR = 10\, log_{10} \left[\frac{\sum_{n=1}^{N} v(x_n)^2\, J_\phi(x_n)}{\sum_{n=1}^{N} (v(x_n) - g(x_n))^2\, J_\phi(x_n)} \right]. \tag{4}$$

Where $v(x)$ and $g(x)$ are the foveated image and the image formed on the human eye, respectively. The metric is evaluated against various images with noise appearing the densest and the sparsest close to the foveation point. FNSR can effectively measure this spatially varying additive noise while showing a clear distinction in metric scores. When evaluated against frequency-based noise FNSR is not able to adequately quantify the noise, however.

Fig. 4. Foveated image in Cartesian coordinates (left) and curvilinear coordinates (right) [14]

3.4 FovVideoVDP

More recently, Mantiuk et al. proposed a foveated video quality metric: FovVideoVDP [8]. The method incorporates spatial, temporal, and peripheral aspects simultaneously. Furthermore, it is a *Physically calibrated metric*, meaning that the metric is display-dependent. This is because the paper argues that, due to the diversity of displays found in current years, display-independent is no longer sufficient. FovVideoVDP intends to model the early stages of the HVS as simply as possible while retaining all important aspects of low-level vision. Because the metric is designed to be differentiable, it can also be used as a loss function in a neural network. To obtain the metric, the algorithm passes several stages:

1. The first step is to convert the original image to the display model used in the paper, which is in terms of physical units.

2. The second step addresses the temporal characteristics of the human visual system, by splitting the video up into a sustained

channel for encoding slow changes, and a transient channel for encoding fast changes.

3. The videos are decomposed using a Laplacian pyramid, to mimic the decomposition occurring in the visual cortex.

4. A CSF is applied, which predicts the sensitivity for foveal and peripheral vision as a function of spatial frequency, temporal frequency, background luminance of the display, and size.

5. Contrast masking is accounted for by normalizing the contrast value by a detection threshold to obtain the perceived contrast.

6. Finally, the perceived difference measures are pooled across all spatial frequency bands, temporal channels, and frames. The pooled visual difference is then regressed to obtain a final quality score.

The metric is evaluated extensively and achieved good performance in general. Execution times were evaluated to be substantially shorter than other metrics of similar complexity. FovVideoVDP and a number of other foveated image and video metrics have been tested against 4 different datasets and compared against subjective quality scores and FovVideoVDP performs as best overall metric.

3.5 HDR-VDP2-FOV

Mantiuk et al. [7] propose a non-foveated image and video quality score which predicts visibility and quality. Visibility is defined as the probability that differences between a test and reference image are visible to an average observer. Quality is defined as the quality degradation with respect to the reference image, expressed as a mean-opinion-score. Similar to FovVideoVDP [8], the metric is physically calibrated. Moreover, it takes many features of the HVS into account, such as intra-ocular light scatter, photoreceptor spectral sensitivities, separate rod and cone pathways, contrast sensitivity across the full range of visible luminance, intra and inter-channel contrast masking, and spatial integration.

In [16], Swafford et al. propose an extension, HDR-VDP2-FOV, reducing the outcome of the CSF depending on the cortical magnification factor, that depends on eccentricity from the fixation point. The contrast sensitivity function for HDR-VDP2-FOV is given by

$$CSF(e,M) = CSF(e) - CSF(e) \times \left(1 - \frac{M(e)}{M(0)}\right)^{1+\alpha*(1-S)} \quad (5)$$

Where e is the eccentricity corresponding to a pixel position (x, y), $CSF(e)$ is the CSF at that eccentricity, $M(e)$ is the CMF at that position, and $M(0)$ is the CMF at the center of vision. HDR-VDP2-FOV uses multi-scale decomposition: S is used to increase the sensitivity of contrast as the scale decreases (S being 0.5, 0.25, etc.) to allow the model to remain sensitive to large-scale contrast changes. Lastly, α is a tunable parameter controlling the effect of peripheral sensitivity. HDR-VDP2-FOV has been used to assess the performance of a foveated rendering method that is also proposed by Mantiuk et al. [16], but no comparison with other metrics and human trials have been performed.

3.6 Luminance-Contrast-Aware Foveated Rendering

Most foveated rendering methods use a fixed quality decay with increasing eccentricities. While this can obtain good results, the decay parameter often needs to be set conservatively to limit noticeable decay in worst-case scenarios. Tursun et al. [17] propose Luminance-Contrast-Aware foveated rendering, where the parameters for the foveated rendering quality decay depend on a metric based on luminance and contrast for a particular frame, rather than a fixed parameter. Particularly interesting is that the metric works on a low-resolution version of a frame, allowing the computation of the metric to be less expensive. Figure 5 compares standard foveation to content-aware foveation.

Fig. 5. Ordinary foveated rendering techniques (left) using a fixed quality decay for peripheral vision. While this can be a conservative solution, it does not provide a full computational benefit. [17] performs content-adaptive foveation (right) and relaxes the quality requirements for content for which the sensitivity of the human visual system at large eccentricities degrades faster [17]

.

3.7 Blind Video Quality Assessment

FVQA is a no-reference video quality metric. It takes a statistical approach, based on space-variant natural scene statistics and natural video statistics [6]. It implements a space-variant Gaussian distribution, that depends on the gaze position. Additionally, other foveation-specific features are implemented, but no temporal aspects are considered. FVQA is extensively evaluated with their own 2D and 3D VR foveated video quality databases (LIVE-FBT-FCVR) and compared it to full- and no-reference image and video metrics. FOVQA performs very well, achieving a rank-order correlation of 0.938, the highest among the metrics.

4 Comparison

In the previous section we have described seven metrics for assessing quality for foveated images and videos. We will compare the metrics discussed in section 3 and assess the performance based on three aspects:

- **Computational efficiency** When metrics are frequently evaluated, there is a need for computational efficiency. Methods that take relatively long to compute may not be desirable as they will adversely affect the total train time.

- **Features** Between the metrics there is a wide range of features of the human visual system taken into account. Depending on the particular use case, certain features may or may not be desirable or applicable.

- **Applicability** When comparing foveated image metrics, it is important to consider their applicability to the specific use case or application. Metrics that work well for one type of medium or application may not be suitable for others. Thus, selecting appropriate metrics based on their applicability can help ensure that foveated image processing techniques are evaluated and optimized for their intended use case.

Firstly, we note some general observations about the field of foveated quality metrics. It is not a particularly recent field study, since the first metrics date back to 2001. However, research back then was sparse, since applications of foveated rendering were lacking, because VR was not a fully viable commercial product yet, and the existence of eye-tracking tools was limited. With the recent emergence of VR with integrated eye-tracking tools, starting around 2014, foveated rendering and foveated quality metrics have become a renewed topic of interest for research. Still, research is sparse, since for this paper we have only identified four metrics dating after 2014.

4.1 Computational efficiency

FWQI, proposed by Wang et al. [19], does not mention computational efficiency, but due to its simplicity relative to the other metrics, and since it is an image-only metric, we assume computational efficiency is relatively low. FMSE, proposed by Rimac et al. [11]

Name	Video	Spatial	Temporal	Reference	Approach	Calibration dataset
FWQI [19]	No	Yes	No	full-reference	Correlation	None
FMSE [11]	Yes	Yes	Yes	full-reference	Correlation	Own
FNSR [14]	Yes	Yes	No	full-reference	Correlation	None
FovVideoVDP [8]	Yes	Yes	Yes	full-reference	Psychopathy model	Own
HDR-VDP2-FOV [16]	Yes	Yes	Yes	full-reference	Psychopathy model	Own
Luminance-Contrast-Aware [17]	No	Yes	Yes	full-reference	Psychopathy model	Own
FVQA [6]	Yes	Yes	No	no-reference	Natural scene statistics	Own

Table 1. A comparison of quality metrics, columns indicate the name of the metric, whether a metric is designed for video, whether it takes spatial aspects into account, whether it takes temporal aspects into account, what kind of metric it is, the approach and the dataset the metric is calibrated on.

is more computationally expensive due to the expensive process of calculating an approximation of the retinal velocity between frames. To keep computational complexity low, it utilizes Haar filters with only two coefficients, which is a metric based on entropy. Moreover, FMSE keeps the computational complexity artificially low, by keeping the fixation point in the center, which is not representative of a real-world application with eye tracking or object of interest detection. Finally, it concludes that is fairly computationally cheap, but only compares their approach to computationally expensive quality metrics based on entropy. The paper proposing FNSR [14] makes no statements about its computational complexity. However, we can analyze that the computational complexity is quite low since the metric is exclusively based on spatial information. FovVideoVDP [8] is one of the more computationally complex metrics due to its several processing stages. In the paper, a comparison of execution times between different metrics is given, where FovVideoVDP is the slowest among the competitors. However, foveated metrics are not considered in this analysis, and the metrics that are analyzed are of significantly lower complexity. Though, an advantage of FovVideoVDP is that it can be implemented to run on a GPU. The computational complexity of HDR-VDP2, the non-foveated quality metric, is linear in the number of image pixels [7]. Since the computational complexity of the CMF function used for HDR-VDP2-FOV is also linear in the number of pixels, we can conclude that HDR-VDP2-FOV has linear time complexity. For Luminance-Contrast-Aware foveated rendering [17] keeps computational cost low by operating on downsampled inputs. No complexity is given but it has been evaluated that it can calculate the predictor variable by around 1ms for a $\frac{1}{4}$x downsampled 4k image frame. Finally FVQA [6], provides the amount of consumed time for the extraction of spatial, temporal, and quality fall-off features, which take about 2 seconds on average. This time consumption makes the metric unsuitable for real-time foveated VR video quality assessment.

4.2 Features

Table 1. presents a comparison of different features of the quality metrics. FWQI [19] and FNSR [14] are relatively simple metrics, only taking into account cortical magnification. FMSE [11] also considers spatial aspects of human vision by incorporating a reduction in contrast sensitivity, due to motion between video frames. FovVideoVDP [8] is quite extensive, modeling spatial and temporal contrast sensitivity, cortical magnification, and contrast masking, attempting to closely mimic the human visual system. HDR-VDP2-FOV [16], draws from all features incorporated in HDR-VDP2 [11], which accounts for the intra-ocular light scatter, photoreceptor spectral sensitivities, separate rod and cone pathways, contrast sensitivity across the full range of visible luminance, intra- and inter-channel contrast masking, and spatial integration, and adds the CMF to account for foveated vision. Luminance-Contrast-Aware foveated rendering [17] is also based on a psychopathy model which accounts for spatial contrast sensitivity, cortical magnification, and contrast masking to obtain the predictor. Finally, FVQA [6], takes a different approach, being based on natural scene statistics, and implements a CMF on top of that.

4.3 Applicability

For both FWQI [19] and FNSR [14], applicability is limited as temporal aspects are not considered. Therefore for analyzing foveated videos, other metrics incorporating temporal vision are definitely better suitable. Metrics operating exclusively on images also have fewer use cases while real-world applications of foveated images are significantly less compared to foveated videos. FMSE [11] is notably more applicable while it accounts for reduced contrast sensitivity due to motion in a video sequence. FovVideoVDP [8], HDR-VDP2-FOV [16] and Luminance-Contrast-Aware foveated rendering [17] appear to be suitable choices for a wide range of applications in foveated videos, due to many features incorporated in these metrics, as discussed in the previous section. Moreover, due to FovVideoVDP being differentiable, it is possible to use it as a loss function in machine learning models. FVQA [6], is particularly useful when no reference image is available, but its applicability in real-time video quality assessment is limited due to the high computational complexity.

5 Conclusion

The goal of this paper was to contribute to the current state-of-the-art in foveated rendering methods by offering a broad comparison between the currently available foveated image and video quality metrics, and to further determine the metric that is most applicable under different circumstances. Thus, we compared seven image and video quality metrics. We tried to provide a subjective quality assessment based on computational complexity, metric features, and applicability. This comparison was difficult to perform since the metrics were tested on different datasets and hardware. Conducting our own experiments was not possible, since the source code of the metrics was often not publicly available. All in all, we provided an overview of the different metrics and have shown the advantages and disadvantages of each of them.

6 Future work

Although the aim of this paper was to provide a complete overview of the current image and video quality metrics, there is room for improvement. Firstly, we compared the quality metrics on objective and subjective categories, but the metrics we used were only tested on different data sets or were not tested at all. Therefore, comparison of the metrics was often not straightforward. An obvious improvement would be to perform an in-depth comparison of the metrics on the same data sets and compare them with scores given by human test subjects as well. This direct comparison would yield a clearer perspective on the benefits and drawbacks of each approach, and an objective score on the quality of each metric.

Moreover, while research in this area is still scarce, we have tried to give a full overview of all currently available foveated metrics. Of course, it is possible that still some (novel) metrics have been overlooked. A potential improvement of this paper is further analysis of other methods.

Acknowledgements

The authors wish to thank Cara Tursun for reviewing this paper.

REFERENCES

[1] R. Dougherty, V. Koch, A. Brewer, B. Fischer, J. Modersitzki, and B. Wandell. Visual field representations and locations of visual areas v1/2/3 in human visual cortex. *Journal of vision*, 3:586–98, 02 2003.

[2] J. Foley. Human luminance pattern-vision mechanisms: Masking experiments require a new model. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 11:1710–9, 07 1994.

[3] W. S. Geisler and J. S. Perry. Real-time foveated multiresolution system for low-bandwidth video communication. In *Electronic imaging*, 1998.

[4] M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44:800 – 801, 02 2008.

[5] J. C. Horton and W. F. Hoyt. The Representation of the Visual Field in Human Striate Cortex: A Revision of the Classic Holmes Map. *Archives of Ophthalmology*, 109(6):816–824, 06 1991.

[6] Y. Jin, A. Patney, R. Webb, and A. Bovik. Fovqa: Blind foveated video quality assessment. 06 2021.

[7] R. Mantiuk, K. Kim, A. Rempel, and W. Heidrich. Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Trans. Graph.*, 30:40, 07 2011.

[8] R. K. Mantiuk, G. Denes, A. Chapiro, A. Kaplanyan, G. Rufo, R. Bachy, T. Lian, and A. Patney. Fovvideovdp: A visible difference predictor for wide field-of-view video. *ACM Trans. Graph.*, 40(4), jul 2021.

[9] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[10] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

[11] S. Rimac-Drlje, M. Vranjes, and D. Žagar. Foveated mean squared error - a novel video quality metric. *Multimedia Tools and Applications*, 49:425–445, 09 2010.

[12] J. M. Rovamo and V. Virsu. An estimation and application of the human cortical magnification factor. *Experimental Brain Research*, 37:495–510, 2004.

[13] T. Samajdar and M. I. Quraishi. Analysis and evaluation of image quality metrics. In J. K. Mandal, S. C. Satapathy, M. Kumar Sanyal, P. P. Sarkar, and A. Mukhopadhyay, editors, *Information Systems Design and Intelligent Applications*, pages 369–378, New Delhi, 2015. Springer India.

[14] P. Sanghoon Lee, Marios S and A. C. Bovik. Foveated video quality assessmen. *IEEE Transactions on multimedia*, 4:129–132, 03 2022.

[15] R. Soundararajan and A. C. Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):684–694, 2013.

[16] N. T. Swafford, J. A. Iglesias-Guitian, C. Koniaris, B. Moon, D. Cosker, and K. Mitchell. User, metric, and computational evaluation of foveated rendering methods. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '16, page 7–14, New York, NY, USA, 2016. Association for Computing Machinery.

[17] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk. Luminance-contrast-aware foveated rendering. *ACM Trans. Graph.*, 38(4), jul 2019.

[18] Z. Wang and A. Bovik. Embedded foveation image coding. *Image Processing, IEEE Transactions on*, 10:1397 – 1410, 11 2001.

[19] Z. Wang, A. Bovik, and L. Lu. Foveated wavelet image quality index. *Proceedings of SPIE - The International Society for Optical Engineering*, 4472, 10 2003.

[20] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[21] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.

A Survey of Time Step Selection Methods for Scientific Visualization

Martijn Westra, Giouri Kilinkaridis

Abstract— This survey reviews the different approaches used to select the key time steps in spatio-temporal data, exploring their advantages and disadvantages. We also note that use cases like video analysis, environmental monitoring, chemical reactions and even medical and financial data observation vary to the degree that shifts the relevance of the advantages and disadvantages. We provide a quick overview of four proposed methods that we choose to investigate, where we outline the strategy used. These methods are Dynamic Time Warping (DTW), Flow-based selection, Information-Theoretic storyboard and Deep Learning-based selection. We find that the DTW approach is generic in the sense that it leaves responsibility with the user for quantifying the change between time steps. The Flow-based and Information-Theoretic approach are each geared towards a specific visualization goal, those being composite rendering and storyboard view respectively, with a specially fitting and intuitive method of quantifying dissimilarity. The deep learning approach is more of a black box type solution that lets a neural network decide how to quantify dissimilarity, via dimensionality reduction, which in itself spawns another method of visualization via further reduction to 2D. In general we conclude that the nature and characteristics of the data and the research goals intended for the data are to be considered when choosing a method.

Index Terms—Time step selection, spatio-temporal fields.

1 Introduction

Time step selection is becoming increasingly more relevant as spatio-temporal fields studied in the field of scientific visualization are getting larger. Advances in computational capabilities of modern computers enable the use of higher resolutions in both the spatial and temporal dimensions when capturing or simulating time-dependent spatial scenarios. This creates a significantly challenging problem for the scientist to tackle when trying to analyze such large data. The full set of data may be well over the available in-house memory of the hardware that is used, or even the disk space may be short for storing all the data. In order to work with this data efficiently, a subset of the time steps has to be selected. A representative subset is needed in order to gain the most insight into the temporal progression within the volumetric state. This raises the question of which time steps are the most representative, or "salient".

Regular selection is often unsatisfactory due to varying speed and significance of events within the simulation. Important events could be entirely missed out upon while empty, duplicate, or otherwise non-interesting states are included in the selection. Manual inspection of all time steps depends on loading and rendering large amounts of data for visual inspection, taking valuable time away from the scientist to perform a mundane task. An automated solution is needed and several have been proposed that are more convoluted. It is unclear at first sight what is the benefit of one solution over another and what are they key aspects that motivate the choice for a particular method.

In this survey we concern ourselves with investigating and reviewing different approaches that have been proposed. In section 2 we elaborate on the background of spatio-temporal field analysis and time step selection. In section 3 the authors use the overall minimum dissimilarity cost between the original sequence and a subset to determine the best subset that will hold the key time steps. In section 4 we summarize the Flow-Based approach, the authors utilize the flow topology of the dataset to identify important time steps, in section 5 a storyboard is used to represent the selected time steps by calculating the Information Difference between the range of time steps. In section 6 a deep learning approach is proposed using a neural network autoencoder to process the data and generate the key time steps. In section 7 we summarize and review the proposed solutions. Lastly in section 9 we state our general conclusions.

- *Martijn Westra is with University of Groningen, E-mail: m.r.westra.1@student.rug.nl.*
- *Giouri Kilinkaridis is with University of Groningen, Inc., E-mail: g.kilinkaridis@student.rug.nl.*

2 Background

Spatio-temporal field data is often obtained via spatial simulations or measurements. Elements in the time series often consist of 3D arrays of scalar values. Sometimes they can be multivariate in the sense that a single position in time and space maps to an array or a matrix (tensor) of values. Deciding on the best way to render the volume at a single time step is already a challenge of its own. Evidently, rendering a 3D block of opaque voxels is unlikely to be the best option. More often we see techniques based on raycasting and mapping data points to color and opacity to get the most information out of the rendering.

A selection of salient time steps can assist in deciding on an effective method to render single time steps. The change in the state between the selected time steps should then be easily visible and interpretable from the chosen method of visualization. A selection of time steps and the renderings thereof can act as a summary of the simulation which is then easily conveyed through a static medium such as a research paper. In the case of ensemble data containing several time series as a result of parameter variations, time step selection can be used for each of the resulting time series, upon which interesting patterns can quickly be spotted and investigated.

3 Dynamic time warping approach

Dynamic Time Warping (DTW) is a process of optimally lining up two time series. It measures the "cost" of this alignment and it requires a distance metric to compare the state between two time steps. Tong et al. base their proposed technique on DTW, mapping the time series onto a subset of itself. An illustration of this is shown in Figure 1.

Considering two time sequences T and R, if we would like to align them and create a mapping, problems in terms of efficiency appear. The key aspect is to use a dissimilarity function between two data points t_i and r_i, where $D(t_i,r_i)$ is defined to measure the difference between the two elements with the overall minimum Dynamic Time Warping cost being the sum of all the costs for every pair $D(t_i,r_i)$.

The mapping between the elements of T and R is bi-directional as can be seen in Figure 1. Some elements from both sets can be mapped to multiple elements of the other set. This is because of the non-linear nature of the time sequences. To achieve appropriate time step selection from a time sequence T, we need to select the best set of k time steps from T and generate the subset R that is the most similar to T and can be aligned with the sequence T.

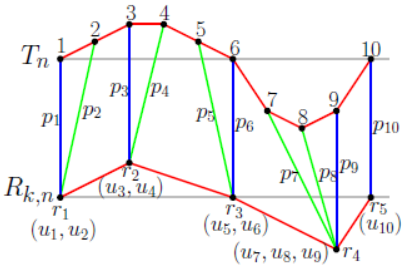


Fig. 1. Illustration of dynamic time warping to a subset of the time steps. Image taken from [4].

As mentioned earlier, efficiency is an issue that can be solved by introducing three constraints to the method.

- The first constraint is to map each time step in the original time sequence T to one and only one key time step in the set R.
- The second constraint states that it makes no sense for a time step not to be representable of itself, that is why any key time step r_i in the selected sequence should be mapped to itself in the original time sequence.
- The last constraint states that the last time step in the original time sequence is always used as a key time step, but this could be a non desired behaviour depending on the data or the purpose of the analysis, that is why it is also possible to introduce one artificial time step as the last one in the original sequence that is completely different from the rest. In the end it is possible to select one more key time step from the total number of $n+1$ time steps, and discard this artificial step after the set R is generated.

Since the problem of obtaining the k key time steps requires multiple computations of the same problem by calculating the minimum dissimilarity cost between two steps, dynamic programming is used allowing to divide the problem into sub-problems, and solve each of the sub-problems in the same way as the original problem until a termination condition is reached. They show that given an optimal selection of time steps, subsets are also optimal, which enables segmented mapping with dynamic programming. They also use this in their visualization app to let the user request more time steps when a particular region catches their interest.

To make it easier for the user to explore the results of the analysis of time-varying data sets, a data browser was designed Figure 2. In the data browser the user can see the selected key time steps and the mapping that was performed between the original and generated sequence. It is possible to navigate time-varying data at different levels of temporal detail. Another feature is that the browser displays information related to the nature of the time-varying data, like the dissimilarity matrix, the key time steps, the jump time steps, and the warping path between the full sequence and the key sequence. The user can interactively choose across the whole sequence the desired number of key time steps or even within a smaller time segment.

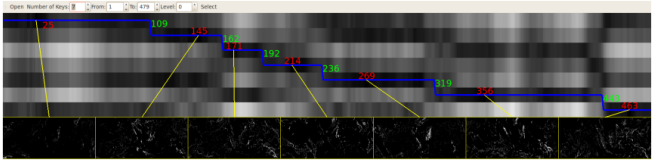


Fig. 2. Time-varying data browser for 7 key time steps. Image taken from [4].

As an advantage they mention that their method does not depend on any particular method of measuring dissimilarity between time steps. The user can decide on which method to plug in.

What they already mention as a limitation is that the method requires computing a dissimilarity matrix and this takes a lot of time using the EMD metric. We have seen methods that try to cut down on this sort of costly computations via probability or sampling methods [1] or other optimization and approximation [6].

They show two applications of their method. One is with a weather phenomenon, Madden-Julian Oscillation. They extract a "Hovmoller diagram" from data where the X axis represents longitude and the Y axis represents time. Then they use the Earth Mover's Distance (EMD) metric with this to compute dissimilarity. In the second example they use isosurfaces for an astrophysics turbulence data set with an isosurface dissimilarity metric. In both cases it appears that the metric used is quite specific to the use case with the data.

4 Flow-based approach

Frey and Ertl propose a flow-based solution aimed to quantify distance in the state of the volume from one time step to another [1]. Here, flow is understood to be the movement of material from one state to another. We illustrate this in a simple 2D scenario in Figure 3.

Frey and Ertl base their method on the *Earth Mover's Distance* (EMD). Otherwise known as the *Wasserstein* metric, this method computes the minimum cost of transforming one distribution of mass into the other, factoring in distance moved for each unit of mass. This means it will compute the minimum cost of the flow in the state between time steps and hence it is a measure of distance in the state at two given time steps.

The EMD suffers from a high computational complexity as a result of using a flow graph connecting every point in the source distribution to every point in the target distribution. To cut down on the number edges in the graph, they propose to use Delaunay triangulation. This is a technique that disallows edges whose circumference would encapsulate another node.

They provide a time step selection algorithm capable of working with progressively loaded data or even streaming data potentially from a simulation that is running concurrently. In the context of the large data that we have in mind, this is a particularly important feature. The selection procedure aims to minimize the distance from the selected time steps to the surrounding time steps from the full set in order to have the most representative time steps.

The underlying assumption behind the flow-based approach seems to be that the subjected data can be modelled meaningfully via such a flow representation. That is, the individual units of material have a start point in time step t and an end point in time step $t+i$, for all possible t, i, and the distance travelled carries meaning that scales, i.e. n times the distance travelled is n times as meaningful. While

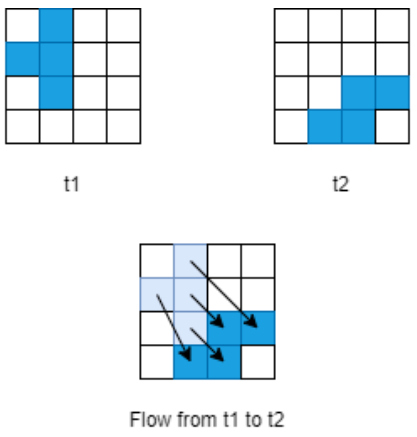


Fig. 3. Illustration showing flow of material in the state of a simple 2D scenario between time steps $t1$ and $t2$

demonstrably excellent when this intuition holds, we raise concerns when using flow-based methods when the intuition does not or only partially holds. Consider an example in the extreme case where objects are formed in-place (without movement) into an empty scene. A distance-based method may have trouble capturing this better than alternative methods.

In their results, they show combined renderings of composited time steps for several data sets and they show that this works best with the flow-based method (as opposed to element-based). This makes sense because when the material does not move significantly, then such a composite rendering will have too much overlap between time steps. It shows that the chosen method is fitted well to the use case. In Figure 4 an example is shown of such a combined rendering, comparing results for flow-based and element-based distances. This visualization shows the expansion of the "hull" of a supernova.

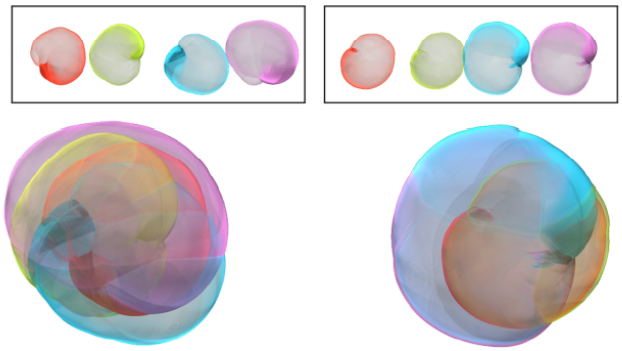


Fig. 4. Composite renderings using flow-based distance (left) and element-based distance (right) of four time steps for the 'hull' of a supernova dataset. Image taken from [1].

5 Information-theoretic storyboard approach

A storyboard in the context of scientific visualization is a time series of images or renderings presented in sequence in a story-telling manner similar to a comic book. Used with a selection of key frames, this can provide an informative overview of the progression within a spatio-temporal field. Intuitively the state between one such time step and the next should be interpretable from the given selection. Zhou et al. propose a selection method basing fitness on similarity in intermediate states compared to interpolations of the nearest selected time steps [6]. An example of a storyboard view is shown in Figure 5.

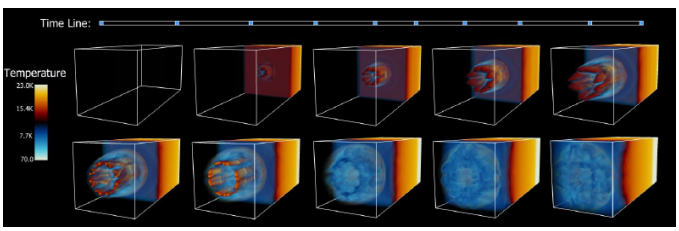


Fig. 5. Storyboard view summarizing temperature in a volumetric distribution. Further details not given. Image taken from [6].

To select representative time steps the authors propose that if for two selected time steps i and j, the skipped time steps in the range (i, j) can be reconstructed from i and j using linear interpolation with a small percentage of missing information compared to the original data, the selected i,j are representative time steps. To achieve this selection a fully automatic dynamic programming algorithm was constructed. The metric that is used to determine if the time steps (i, j) are representative is Information Difference (InfoD) defined as $c(i, j)$, $c(i, j)$ represents the sum of all VI (variation information) of all reconstructed skipped time steps with the original between i and j.

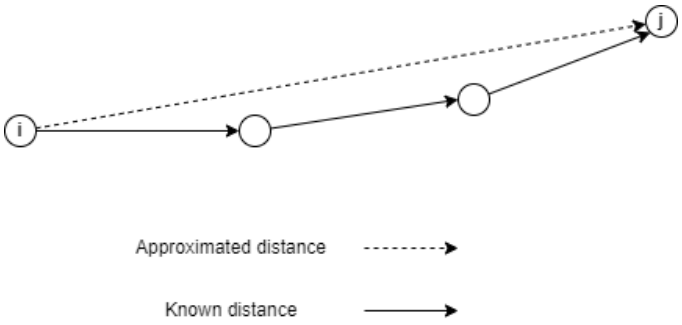


Fig. 6. Illustration showing a distance approximation using in-between known distances

The complexity of computing the table containing all the $c(i, j)$ is $O(T^3N)$ but also the minimum InfoD for every pair must be computed. Since computing the InfoD for every pair is a recurrence dynamic programming can be used to memorizing all the valid tuples as states in a table offering a $O(T^3)$ time complexity, making the over all combined complexity $O(T^3N + T^3)$, where N is the amount of data per time step and T the total number of time steps.

Another concern is the size of the data. If the data can not fit the disk, the I/O operations complexity is $O(T^3(N/B))$ where B is the number of items fitting in one disk block. To solve the running time complexity bottleneck and the high cost of I/O operations the authors created an approximation technique.

Approximation The intuition is that instead of computing for very far apart tuples of i and j the information difference, the $c(i, j)$ value can be estimated from the already computed ones. This is illustrated in Figure 6.

The idea is to use a multi-pass sliding window technique over the set S containing all of the time steps to make it fit in-core $|S| < t$ where t is the number of time steps to fit in-core. On every pass the value of S is halved forming a geometric series and in each pass only the best k (minimum number of time steps) representatives selected using DP and $|S|/2$ are put in S for the next pass, reducing the overall time complexity to $O(t^2TN + T^3)$ and for the I/O to $O(T(N/B))$.

6 Deep learning approach

An autoencoder is a neural network that learns to encode data to a smaller latent representation and then decode to reconstruct the original. Related work has been done in [2] where the authors proposed a method sampling small patches from (Computational Fluid Dynamics) CFD data, training a Siamese deep neural network which has similar structure with two Convolutional Neural Networks (CNN) that selects the key time steps according to the similarities between consecutive time steps which are assessed by the networks.

The authors of [3] propose training an autoencoder-based deep learning pipeline for dimensionality reduction and then proceeding with time step selection in the reduced space. The deep learning approach as described offers the advantage of handling time-varying multivariate data sets simply by changing the number of channels in the autoencoder. This is in contrast to running the complete procedure once for each dimension and then figuring out a suitable way to combine the results. A schematic of the pipeline is shown in Figure 7.

The technique involves two phases: a) feature learning and b) time step selection.

Feature learning To learn the features from the dataset the network has an encoder and a decoder. First the encoder was designed to accept volumes of data in the form $C \times L \times W \times H$. Here, C refers to the number of channels of the volume, where $C = 1$ is for univariate data and $C \geq 2$ for multivariate data. L represents the number of time steps, W the number of time steps considered at once for selecting representative time steps and H represents the number of variables in the data.

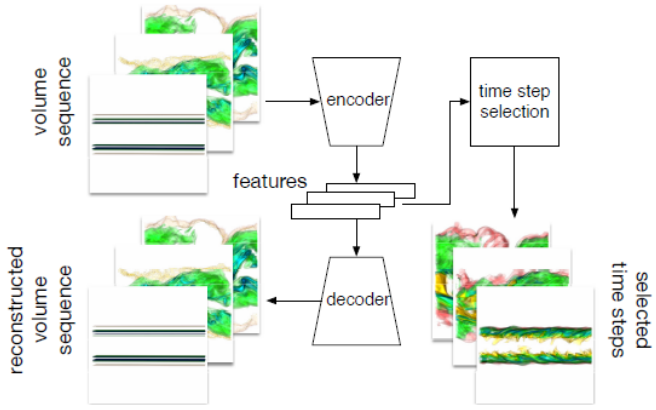


Fig. 7. Schematic of the autoencoder-based time step selection. Image taken from [3].

The encoder generates a feature descriptor for each time step, each feature descriptor has 1024-dimensions. The purpose of the decoder is to reconstruct the original data from the feature descriptors. Then they use the mean square error (MSE) to calculate the loss between the ground-truth and reconstructed volumes. To select the representative time steps, the authors project the feature descriptors in a 2D space via t-SNE [5]. The argument for using t-SNE is that it preserves neighborhoods. They further argue that reducing to 2D makes the visualization match up with the selection and reduces the complexity of distance computation, as opposed to selecting in higher dimensional feature space.

An example of how they display the temporal development of a simulation using t-SNE projections is shown in Figure 8.

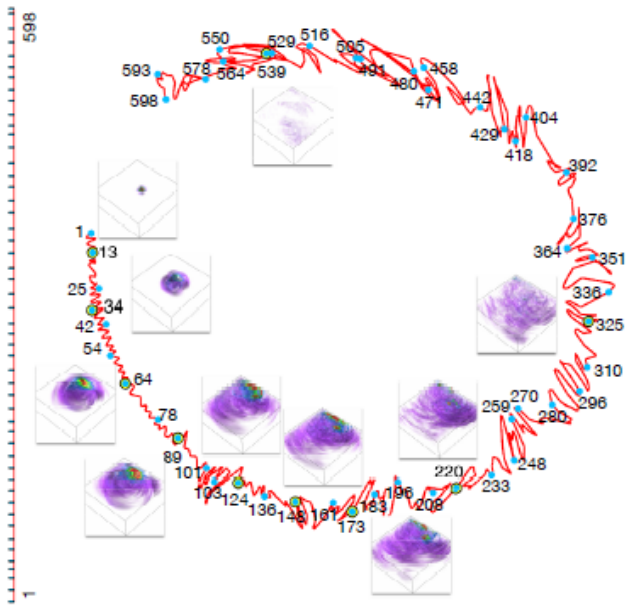


Fig. 8. t-SNE projection showing temporal progression of a vortex data set. Image taken from [3].

Time step selection When all the feature descriptors are generated and reduced to 2 dimensions and projected on a 2D space, where each point represents a time step, the neighboring time steps are connected to form a path. To select the representative time steps from that path 3 methods can be used arclength-based selection where the Euclidean distance is computed between the neighboring points, Angle-based selection where angle formed among consecutive neighboring points is computed based on a angle threshold θ and Mixed selection where the two previous methods are combined where a threshold $\alpha \in [0,1]$ is used to manipulate the importance between arclength and angle.

7 Discussion

Tasked with reviewing the different approaches to weight advantages and disadvantages, we find that there is plenty to talk about but we also find that use cases and data vary to the degree that shifts the relevance of these (dis)advantages. We cannot speak for the user and what they want in their specific scenario so we give a broader outline on the key aspects to review when choosing a method.

7.1 Dynamic Time Warping

For the Dynamic Time Warping approach [4] the advantages are that the generated subset containing the key steps is selected using the globally optimal minimum dissimilarity cost. The technique over all is not bound to the use of one specific dissimilarity metric, based on the needs of the user EMD or K-L divergence can also be used. On the other hand as the authors state they need to use the entire sequence to compute the dissimilarity matrix, which is time consuming. Also if the user doesn't need the last time step as a key step he has to prepossess the data and create manually an artificial time step and add it to the end of the sequence.

The user is responsible to figure out what dissimilarity measure he needs to obtain the key time steps depending on his data. The method as-is works with precomputing a dissimilarity map which looks like it could benefit from some sort of caching solution to save on this computation in cases of expensive dissimilarity metrics.

If the user already has decided on the dissimilarity metric with which to quantify the change between time steps, and the precomputation of the dissimilarity map does not cause any problems when used with the data they have in mind, then the DTW solution is an excellent choice.

7.2 Flow-based

The Flow-based approach [1] is built around the EMD metric, based on quantifying movement of material, offers faster computation and high accuracy. This works very well with the composite rendering of multiple time steps in a single visualization because time steps will be selected when the material has moved a significant amount, which will make it so that the volume renderings do not occlude each other too much.

However, it may not necessarily always be the best way to quantify dissimilarity by using the movement of the material. In some cases the importance of an event may not be scaling with the moved distance. Further, there may be additional material introduced into the scene across time steps that did not originate from somewhere in the scene, such that the EMD cannot quantify this change. The assumption is that between time steps all units of material have a start and end point, which is not necessarily always true.

The method excels in cases where movement of material is to be quantified and the relation to time is to be shown in the selection of the time steps, or the amount of time that has passed. The provided scheme yielding multiple selections of an increasing amount of time steps, combined with the visualization, allows the user to see the progression and opt for a selection that catches their interest.

7.3 Information-Theoretic

The information-theoretic storyboard method [6] is specifically geared towards the storyboard visualization, basing their method on the intuition that the state in between selected time steps should resemble interpolations of the selected time steps. This makes it so that the user knows what to expect from the visualization. We raise concerns for the cases where this intuition does not work well, such as highly nonlinear events in the simulation.

The paper introducing this method is rather short. It does not highlight the advantages and usages to the same degree as some of the other

papers we looked into. The dynamic programming solution should be able to offer the same benefits for interactive visualization as the DTW solution did.

7.4 Deep learning

The deep learning approach [3] offers a widely applicable solution. We praise its wide and general applicability but we also raise concerns at the black box nature and lack of intuitive underlying principles guiding the selection that would yield predictable results. Rather, this sort of approach seems to let a neural network decide what constitutes a significant change in the state by forcing it to reduce in dimensionality.

The results are good but perhaps not necessarily fit for a other visualization goals. The t-SNE projections given are adequate visualizations in themselves for displaying the temporal behaviour. However, we are unconvinced that the proposed solution transfers well to other use cases that we have seen. The deep learning approach has to be trained and is specific to the data. We are unsure about the option to compare results across data sets for that reason.

The method is advertised with the advantage of supporting multivariate data simply by adding more channels to the autoencoder. We are unsure how this relates to the other metrics that are applied directly to the data, in which case the output could be use in a vector sum or magnitude to augment support for multivariate data.

For those individuals with little to no information on the general temporal behaviour of the data set that they are interested in, those individuals looking to do exploratory analysis, or those who may have multivariate volumes, we recommend this technique. The authors themselves do not appear to state much in regards to limitations and future work. They only say that options exist in the direction of reinforcement learning. This appears to be one of the newer works. We would be interested to see the direct comparison to flow-based selection in the context of composite rendering as well as the comparison to the information theoretic approach in a side-by-side storyboard view.

8 Limitations and Future Work

In our survey we mainly focus on the intuition and the theoretical aspect of the chosen methods for quantifying dissimilarity between time steps because these metrics tends to be the main focus in the works we investigated. The respective procedures for selecting time steps and visualizing the selection often depend (to some degree) on the chosen metric, so they are more difficult to compare directly. It would be interesting to see experimental results showing to what extend we get different results when using different metrics under the same key frame selector. Further, we would like to know how do the results vary across different datasets. This would give us more insight in the significance of the choice between the metrics and it would extend the theoretical basis we already have for guiding this choice. Additionally, it may be of interest to see the effect of different key frame selectors, as well as the result of different methods of visualization in these different experimental settings.

9 Conclusion

We cannot say that one method is strictly better than the other. We point the scientist looking for a suitable time step selection method to be conscious of the kind of data they have and the research and visualization goals they have, so that they can verify that the chosen method aligns with these.

A common problem that all methods face is the time complexity and computational efficiency, especially with larger data. To get representation of a high-dimensional simulation, selecting salient time steps is a must. If the amount of selected key time steps is too large, there could be redundant information or even noise in the data. Making it quite hard to identify patterns or trends that lead to meaningful conclusions. Selecting too few steps and important information is being overlooked and lost. Since the real world phenomena and simulations produce large amounts of data, analyzing them is a computationally intensive task. Partitioning data and processing it in parallel is not an easy task. The authors of the proposed techniques mention that they need many hours and decent hardware equipment to generate results. In most occasions the authors had to find a balance between accuracy and efficiency.

The complexity of the proposed methods and the amount of research required to decide on the most suitable method for one's own scenario can seem daunting. We hope that our discussion can help shed some light on the general strategy, the key aspects, and the advantages and disadvantages of the different solutions that are proposed, so that the interested scientist working with spatio-temporal fields can have a quick start and focus on the important characteristics of their specific dataset.

Acknowledgements

The authors wish to thank expert reviewer Dr. Steffen Frey, and those who will peer review

References

[1] S. Frey and T. Ertl. Flow-based temporal selection for interactive volume visualization. In *Computer Graphics Forum*, volume 36, pages 153–165. Wiley Online Library, 2017.

[2] Y. Liu, Y. Lu, Y. Wang, D. Sun, L. Deng, Y. Wan, and F. Wang. Key time steps selection for cfd data based on deep metric learning. *Computers and Fluids*, 195:104318, 2019.

[3] W. P. Porter, Y. Xing, B. R. von Ohlen, J. Han, and C. Wang. A deep learning approach to selecting representative time steps for time-varying multivariate data. In *2019 IEEE Visualization Conference (VIS)*, pages 1–5. IEEE, 2019.

[4] X. Tong, T.-Y. Lee, and H.-W. Shen. Salient time steps selection from large scale time-varying data sets with dynamic time warping. In *IEEE symposium on large data analysis and visualization (LDAV)*, pages 49–56. IEEE, 2012.

[5] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9(nov):2579–2605, 2008. Pagination: 27.

[6] B. Zhou and Y.-J. Chiang. Key time steps selection for large-scale time-varying volume datasets using an information-theoretic storyboard. In *Computer Graphics Forum*, volume 37, pages 37–49. Wiley Online Library, 2018.

Fairness in Software Teams: Challenges and Solutions

Tom Eijkelenkamp

Abstract—We give an overview of some of the studies describing fairness in software engineering. Diversity of a team can improve fairness and bias is seen as unfair. We show data on diversity in teams on a global scale as well as how it evolves over time. We show data on performance between women and men. In addition to this, bias that shows up in terms of gender and geolocation and provide data of studies with possible solutions to bias.

Index Terms—fairness, diversity, bias, gender, geolocation.

1 Introduction

There are studies [11] that show that variety is important for a team to improve productivity and also gender diversity will make an environment that is more fair [10]. But then this diversity is not seen in many places in the world. Women make up less than 10 percent of core contributors [1]. One study suggests that there is a high increase in new jobs in the software engineering sector, because of the revolution in electronics, this job demand could be solved by encouraging women to enter the software industry [9]. The European Commission estimated that if more women would join the job sector of the digital world, it could create an annual EUR 16 billion GDP boost for the European economy [2]. Numbers show an increase in the fraction of developers that are women, but this is not much and could be much more.

Fig. 1. Gender diversity in 2014 of open source projects on GitHub in according to [6]. Darker regions show higher diversity.

2 Study

First we will show some data on the current state of diversity in the world and what elements are of influence for developers to work and continue working on a project. Then we show some data that cast doubt on a hypnosis that minorities might perform less than the average developer. In addition to this we show that there are influences of bias regarding minorities and evaluate possible solutions to these unfair behavior.

2.1 Diversity around the world

The paper [6] goes deeper into all the numbers of this diversity across the world. They use data from GitHub open source software projects to get an overview in the participation and look into if there is a substantial difference between regions. When we see some countries do better than others, we could look into why this is the case and learn how we can improve the diversity in other countries. Although the data from the study does not show a statistical difference in diversity among all the regions in the world, the overall participation of women is very low. Still we could say that the highest diversity exists in Asia and the Americas. This was not significant, but the increase over time is. They show how this maps out over the world, shown in fig. 1 and fig. 2.

Fig. 2. Gender diversity in 2023 of open source projects on GitHub in according to [6].

To improve the whole situation of diversity the study [6] goes on in asking developers what they find important for participating on a project. When asked these questions, the researchers find that the most important aspects in starting or continuing on a project is that people would regard goal alignment as most important, as well as a welcoming community. Continuation of existing tasks and low stress levels will make developers stay on the project. The developers responded very low towards promotion on social media, and fairly low on friends or colleges that happen to be on the same project.

The reasons for leaving a project are diverse. The developers responded with answers as lack of time or money, there are better opportunities elsewhere, the project is inactive so there is no reason to participate, the engineering environment is poor in terms of a complex installation process or architectural structure, there is no proper roadmap or clear README, also other reasons for people to no con-

• *Tom Eijkelenkamp is a student at the University of Groningen, E-mail: t.eijkelenkamp@student.rug.nl.*

tinue a project is that the community is toxic or discriminating.

So that sketches an idea overall of what people find important for working in an open source software environment. What is specific for women is that they regard working on a team with same gender colleges as important as opposed to men who don't see it as important to work along with other men. Also for women it's more important to work with friends or colleagues, and getting paid for the job they do. For the developers worldwide to contribute to projects that consist of team members speaking a different language, the women find it a bit more helpful than men to have good translation tools. When the developers were asked what could increase the number of women in the field, they responded that making the people aware that there are other women working there would increase the likelihood of more women wanting to participate. In addition to this they would want to see women in higher positions, as role models in the field. The study goes on where they believe that creating an automated system for assigning exciting/challenging tasks among developers to motivate continued participation of underrepresented communities.

The motivation for staying on a project differs from region, for example in Afrika it's much more important to get paid, relative to Europe, Asia or America. Exciting tasks are wanted more in Asia and Africa, although they are wanted everywhere. Connection to people worldwide is not so much a factor for people from Europe or America.

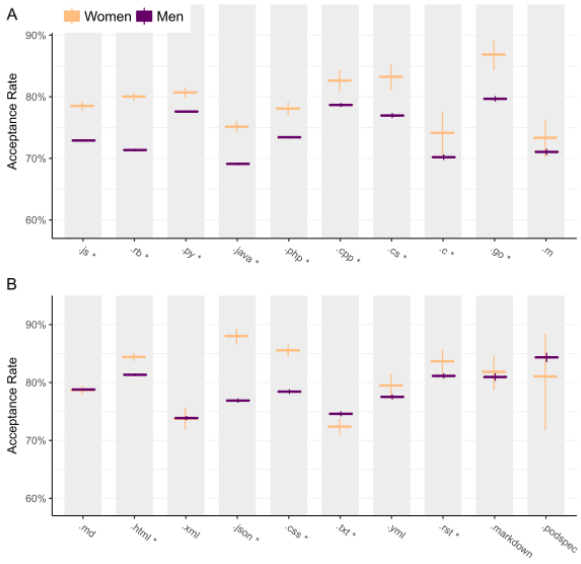


Fig. 4. Ratio accepted merge-request per language, women versus men [10].

2.2 Code written by women is more often accepted

Another study [10] shows that women outperform men when we compare their contributions to open source software in terms of the ratio of contributions getting accepted. This is study again is performed on the data of Github, the open source in specific. The projects of github can work in two ways, one way is that developers can contribute to projects directly, commit changes to branches to add features to the product. Another way is to create merge-requests, here a contributor requests a review for a feature that he or she wants to add. The project owners evaluate the new code and either accept it or reject the merge.

When looking at the data of all the merge-requests done in GitHub (specifically filtered to obtain the relevant merge-requests), it shows that the contributions of women are accepted more often than men. Code provided by women is accepted 78.1% of the time, while for men this is 74.1%, exact numbers shown in Fig. 3.

Gender	Open	Closed	Merged	**Merge Rate**	95% Confidence Interval
Women	1,573	7,669	32,944	**78.1%**	[77.69%,78.49%]
Men	60,476	297,968	1,023,497	**74.1%**	[73.99%,74.14%]

Fig. 3. Ratio accepted contributions per gender of open source projects on GitHub in according to [10].

This might suggest that women write better quality code and they are more wanted in the field of software development. A paper [4] suggests that developers are wanting to help women gain a position in the field.

The research [10] goes on why the code is accepted more often, looking at the statistics of the contributions done. They find that women do not use simpler code languages, the overall diversity of used languages is similar to men. The statistics on this are shown in Fig. 4.

As found in another study [3], the number of code lines changed negatively influences the change of a contribution to get accepted, but the data shows that women on average change more lines of code. So it's not the case that updates made by women are smaller, so accepted more often, actually the opposite is true. Another statistic they look at is the number of projects that are contributed to, but again this is not providing any insight into why merge-requests are accepted more often for women.

An explanation could be that the acceptance rate of women starts lower and increases over time, due to the survivorship bias [8]. The women who remain in the field of software development will be equipped to defend their contributions, but at the start be more likely to be judged by gender bias and be less likely to get accepted. Although the data does not show this, comparing the ratio of accepted contributions over time from 1 to 64 contributions, the code provided by women is accepted more often. This casts doubt on the hypothesis that women have to use an aggressive argument style to justify one's own contributions. Fig. 5 shows the ratio of contributions that get accepted starting at someones first contribution to every succeeding.

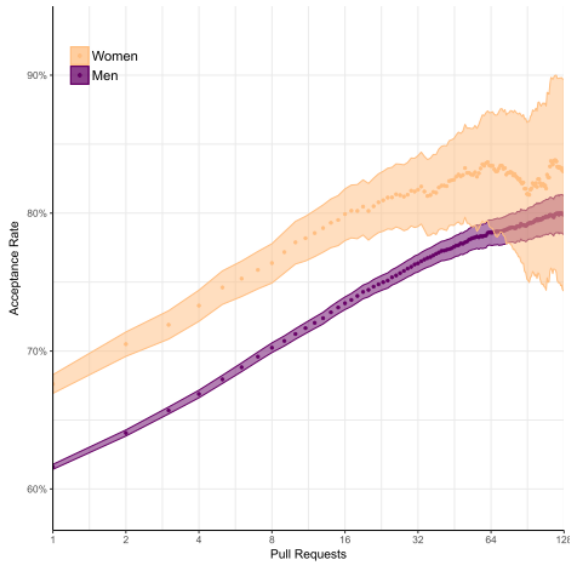


Fig. 5. Ratio accepted merge-request over time [10].

2.3 Bias in OSS

Although we see the contributions of women accepted more often, this ratio goes down when the contributions are from women that are outsiders to the project and the gender is shown or can be guessed by looking at the profile information. When it can be guessed what the gender of an outside contributor is, the ratio of accepted merge-requests is lower compared to men. The ratio for outside female contributions drops from 70% when gender is not visible to 58% when gender type is visible. Also for men there is a drop from 65% to 61%, from gender-neutral profiles to gendered, but this drop is smaller. Fig. 6 shows this in a figure.

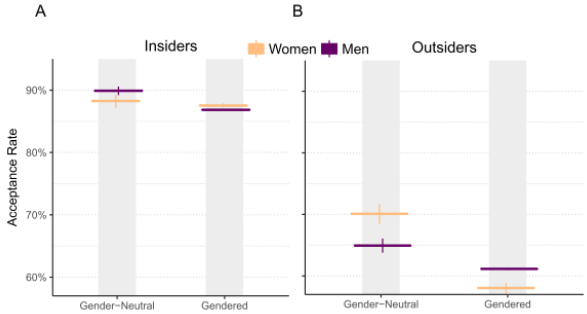


Fig. 6. Ratio accepted merge-request, gender is known versus gender is not known. [10].

The authors [10] of the paper find similar results when controlling the data for covariates. As shown in the previous section women tend to contribute larger code changes than men. A bigger commit will decrease the likelihood of it getting accepted. By controlling the data for this imbalance, a similar drop in acceptance ratio is found due to gender bias.

This bias is only shown in contributions from outsiders and this study involves open source software and not proprietary software. When contributions are from insiders or proprietary software the developers are probably well known in the community, so the gender as well. This makes the comparison for gender-neutral versus gendered not easily possible, making it hard to tell if this bias exists in other sectors.

For developers to start working on a project or continue to be investing it is important to have a welcoming community. Also discrimination and toxic environments are reasons for developers to stop working on a project. This gender bias does not seem to be supportive of the gender diversity of projects.

Another study [7] looks at the bias from the point of view of geolocation. They perform the study on open source software in GitHub as well, where they compare the ratio of contributions that get accepted between countries. Developers can contribute to projects all over the world. A developer from one country can work on a project in another country. This study shows that the ratio of accepted contributions correlates significantly with the geolocation of the contributor. They found that the most accepted countries are Zwitserland, The Netherlands and Japan. Countries such as China, Italy, Brazil and Germany have the lowest acceptance rate. Fig 7 shows the correlation between country and acceptance of a pull request.

	Model 1	Model 2
(Intercept)	2.82 (0.14)***	2.61 (0.14)***
Control variables		
proj_months_existence	−0.01 (0.00)***	−0.01 (0.00)***
proj_watchers	−0.00 (0.00)***	−0.00 (0.00)***
log(proj_ncloc + 1)	−0.06 (0.01)***	−0.06 (0.01)***
proj_external_contribs	−0.01 (0.00)***	−0.01 (0.00)***
proj_test_loc_per_lloc	0.00 (0.00)***	0.00 (0.00)***
log(dev_followers + 1)	0.06 (0.01)***	0.07 (0.01)***
dev_commit_access	0.06 (0.07)	0.05 (0.07)
dev_followed_integrator1	0.11 (0.03)**	0.10 (0.03)**
dev_watched_project1	0.04 (0.03)	0.05 (0.03)
log(dev_prev_pull_requests + 1)	0.17 (0.01)***	0.17 (0.01)***
dev_success_rate	0.01 (0.00)***	0.01 (0.00)***
dev_months_participation	−0.00 (0.00)***	−0.00 (0.00)***
log(pr_comments + 1)	−0.25 (0.01)***	−0.24 (0.01)***
log(pr_changed_loc + 1)	−0.06 (0.01)***	−0.06 (0.01)***
log(pr_changed_files + 1)	0.01 (0.02)	0.01 (0.02)
pr_test_inclusion1	0.26 (0.03)***	0.26 (0.03)***
geo_country_switzerland	0.38 (0.11)***	0.46 (0.11)***
geo_country_netherlands	0.26 (0.09)**	0.36 (0.09)***
geo_country_japan	0.25 (0.08)***	0.34 (0.08)***
geo_country_united kingdom	0.13 (0.04)**	0.20 (0.04)***
geo_country_canada	0.12 (0.07)	0.22 (0.07)**
geo_country_belgium	0.09 (0.12)	0.18 (0.12)
geo_country_spain	0.08 (0.10)	0.15 (0.10)
geo_country_australia	0.05 (0.07)	0.14 (0.07)
geo_country_india	0.02 (0.07)	0.12 (0.07)
geo_country_france	0.02 (0.06)	0.11 (0.06)
geo_country_russia	−0.06 (0.07)	0.04 (0.07)
geo_country_sweden	−0.21 (0.09)*	−0.10 (0.09)
geo_country_germany	−0.25 (0.04)***	−0.16 (0.05)***
geo_country_brazil	−0.27 (0.06)***	−0.19 (0.07)**
geo_country_italy	−0.31 (0.08)***	−0.21 (0.09)*
geo_country_china	−0.39 (0.09)***	−0.27 (0.10)**
geo_same_country1		0.18 (0.03)***
AIC	49231.59	49198.20
BIC	49534.09	49509.87
Log Likelihood	−24582.80	−24565.10
Deviance	49165.59	49130.20
Num. obs.	70740	70740

***$p < 0.001$, **$p < 0.01$, *$p < 0.05$

Fig. 7. Logistic regression models of factors influencing pull request acceptance [7]

When developers do a pull-request on projects that are located in their own country the likelihood of the request to be merged is higher, then requesting provided changes to a project located in a foreign country. This is true for all countries except India shows a lower acceptance rate for contributions from their own countries.

The reasons for why this country bias is there is not clear and could be researched in more depth.

2.4 Possible solutions to bias

A solution to gender or geolocation bias might be to remove this information in the code review process. When a code reviewer would not know the gender or geolocation of a developer, these facts will not influence the decision process of code getting accepted or not.

A trial [5] of such a system is experimented with in Google, where they made a code review application where personal information if the developer is left out. The participants responded neutral or slightly positively towards how this would improve fairness. This study is done on proprietary software, here they find more often than not the author and so personal information can be guessed by for example the style of the code. The participants were mostly neutral towards whether this would change the quality of the code. In order to communicate in person or grant access to certain parts of the project the identity removal of the developer made the process inconvenient to work with.

As earlier discussed to create a more welcoming community, it would help that it's seen by others that there is a diversity of employees. Someone can doubt that this anonymous review system does create an open community in that sense.

A solution for the influence of a knowing that a developer is from a certain country or what type of gender the developer has to reviewing the quality of code might not be simple. Creating awareness for knowing certain information about the author can change the outcome of the reviewing process, may remind people to base there judgment on necessary information.

3 Conclusion

Diversity in a team improves the productivity of a software engineering project. This diversity is low worldwide in the field of software development, but has increased in the past years. To make people attracted to projects a welcoming community is important. For women it is important to have other same gender colleagues and role models in higher positions to join the field. Reasons for joining a project and continuation of contribution varies from region to region. Bias might create barriers to the community, but solutions are not simple.

4 Discussion

The studies described are mostly involving data from open source software projects. The diversity may be different in proprietary software as well as many other statistics. Getting paid for example may be much more important for working for proprietary software, so attracting people to work in those types of settings may need very different approaches. Also the proportion of bias can be different for proprietary software, maybe something as time spend on a project can influence how much gender or nationality bias will make a difference in judgement.

5 Future work

On a future note the research could be extended towards proprietary software. For example evaluate bias is also experienced in situations where the developer has a longer contract and is known to the community. In addition to this we could evaluate if there is an influence of bias for how much salary some developer receives.

References

[1] A. Bosu and K. Z. Sultana. Diversity and inclusion in open source software (oss) projects: Where do we stand? In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–11, 2019.

[2] E. Commission, C. Directorate-General for Communications Networks, and Technology. *Women in the digital age : final report*. Publications Office, 2018.

[3] G. Gousios and A. Zaidman. A dataset for pull-based development research. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, page 368–371, New York, NY, USA, 2014. Association for Computing Machinery.

[4] A. M. Lisa. Spotlighting: Emergent gender bias in undergraduate engineering education. *Journal of Engineering Education*, 94(4):373–381, October 2005.

[5] E. Murphy-Hill, J. Dicker, M. M. Hodges, C. D. Egelman, C. Jaspan, L. Cheng, E. Kammer, B. Holtz, M. A. Jorde, A. K. Dolan, and C. Green. Engineering impacts of anonymous author code review: A field experiment. *IEEE Transactions on Software Engineering*, 48(7):2495–2509, 2022.

[6] G. A. A. Prana, D. Ford, A. Rastogi, D. Lo, R. Purandare, and N. Nagappan. Including everyone, everywhere: Understanding opportunities and challenges of geographic gender-inclusion in oss. *IEEE Transactions on Software Engineering*, 48(9):3394–3409, 2022.

[7] A. Rastogi, N. Nagappan, G. Gousios, and A. van der Hoek. Relationship between geographical location and evaluation of developer contributions in github. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '18, New York, NY, USA, 2018. Association for Computing Machinery.

[8] J. Reagle. "free as in sexist?" free culture and the gender gap. *First Monday*, 18(1), Dec. 2012.

[9] K. Stephens and K. S. Crandall. What twitter is saying about women in technology. In *2022 Intermountain Engineering, Technology and Computing (IETC)*, pages 1–4, 2022.

[10] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, and J. Stallings. Gender differences and bias in open source: pull request acceptance of women versus men. *PeerJ Computer Science*, 3(111), 2017.

[11] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov. Gender and tenure diversity in github teams. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3789–3798, 2015.

What makes a great software team?

Elnur Seyidov and Mike Lucas

Abstract—The formation of a software development team serves as a crucial driving force for significant software projects. Its composition can vary, potentially affecting the project's outcomes in various domains, such as development budget, quality, scope, and timeline. Therefore, it is important to gain a more profound understanding of the factors that impact the formation of an exceptional software development team. The primary objective of this paper is to assess the significance of a great software development team, investigate the factors that impact its creation, and propose strategies for enhancing the team formation process.
To address the aforementioned points, a review of relevant literature was conducted. The study revealed that the development of a highly effective software team could potentially result in a number of benefits within the team. Those benefits include enhanced employee and team performance, and improved team and company reputation. Moreover, the study has also established that the main factor influencing team composition is primarily the technical skills possessed by the potential team members. Lastly, the study explored the potential of incorporating soft skills as a requirement during the team formation process to improve the team's effectiveness.

Index Terms—software development team, personality, soft skills, technical skills, project success.

1 Introduction

The level of success in which a project is finished is determined by a number of factors. These often include factors such as time, cost, and scope. This is known as the triple constraint in project management [3]. In the case of software development, the resources working on a project mostly consist of human individuals. Therefore these triple constraints are influenced by the formation of a team working on a project. Aspects such as productivity, quality, efficiency, and cost more greatly depend on the ability of individuals to communicate and work together. It can also affect the success of the project in aspects such as the development budget, quality, scope, and time. Therefore, it is critical for developing a deeper understanding of the elements that influence the formation of a great software development team, which will benefit project managers.

This research aims to answer what makes a great software team. This question will be answered by combining the answers of three sub-questions regarding the relevance of a great team, what influences the formation, and how it can be improved. The answer to these questions is mainly provided by summarizing 4 papers, see section on paper selection.

The aforementioned sub-questions will constitute the research questions that this paper attempts to answer. The research questions are listed below.

RQ1 What is the relevance of a great software team?
RQ2 What factors influence software team formation?
RQ3 What is done to improve software team formation?

By summarizing the information present in the papers mentioned earlier, these research questions should be answered and provide insight into the bigger picture of what makes a great software team. This information is valuable to project managers who can utilize it for better team formation resulting in more optimized satisfaction of the triple constraints.

1.1 Paper selection

As previously mentioned, this paper attempts to answer three research question based on information primarily found in 4 other research papers. These papers are selected based on the topics/questions they discuss, and its ability to potentially answer the research questions. The papers are briefly discussed now. One mapping study discussing the state of the art of software development team formation [8], mentions concepts such as the algorithms used and relevant data that is used for the process. A literature study that dives deeper into the automation process of software development team formation [20], discussing various algorithms. A paper that considers how personality types and combinations of them influence various aspects of completed projects [14]. And finally, a fourth paper talks about how social ties influence team formation, for open source projects, in particular, [11]. Why do these papers have potential to answer the research questions? Because the mapping study should provide an overview of research and an insight into the current state of affairs regarding team formation. The paper about automation can provide insight into how team formation takes place and how it is improved. The papers about social ties and personality types can give us insight into what factors influence the team formation. The topics of these papers seem promising in their ability to provide answers to the research questions. These 4 papers present the main source of information, however sometimes other papers are also consulted for bits and pieces of specific information.

1.2 Paper structure

The structure of this paper will be discussed now. The following section (2) will cover the background information needed to understand some of the concepts discussed in this paper, regarding factors, soft skills, and hard skills. Then in section (3) research question 1 will be answered about what the relevance is of a great software team. Section (4) will answer research question 2 about what factors influence software team formation. In section (5) research question 3 will be answered covering how the software team formation can be improved. Then in section (6) a short discussion of the findings is discussed, which can hopefully answer the bigger question of what makes a great software team. Finally, in section (7), the paper is concluded.

2 Background

When evaluating the process of team formation in software development it is necessary to assess the employees based on their skills. Two general types present in developers are soft skills and hard skills.

2.1 Soft skills

Soft skills refer to the personality traits and attitudes that drive a person's behavior. [21]
According to [1], 9 soft skills were identified in the area of software development

- *Elnur Seyidov, MSc Student Computing Science at University of Groningen, E-mail: e.seyidov.1@stude.*
- *Mike Lucas, MSc Student Computing Science at University of Groningen, E-mail: m.lucas.1@student.rug.nl*

- **Communication skills**: The capacity to communicate information in a way that is comprehended and accepted effectively
- **Interpersonal skills**: The ability to interact with individuals through social communication and engagements, even in challenging circumstances.
- **Analytical and problem-solving skills**: The capacity to comprehend, express, and resolve intricate problems, and to make reasonable judgments using the information at hand.
- **Team player skills**: An individual who is capable of collaborating efficiently within a team setting and contributing to the achievement of the intended objective.
- **Organizational skills**: The skill to effectively handle multiple responsibilities and adhere to timelines while minimizing resource wastage.
- **Fast learner**: The capacity to grasp novel concepts, techniques, and technologies within a relatively brief period.
- **Ability to work independently**: Able to complete the received assignments with little to no oversight at hand.
- **Innovative**: The capacity to generate innovative and original solutions to any given problem.
- **Open and Adaptable to change**: The capability to acknowledge and accommodate modifications while performing a task, without demonstrating resistance.

In addition to the listed social skills, [14] states that personality types may also be highly regarded when forming a software team. The aforementioned paper references Freud and Adler's study [13], suggesting a system for categorizing people according to their 3 psychological functions, which entails determining their preferences over others. It also completes the technique by later being referenced by Myer-Briggs, who adds to the list a new category [6].

- **(E/I)**: The contrast between *extroversion* and *introversion* is determined by an individual's source of energy, and this contrast is also reflected in their distinct approaches to gathering information.
- **(N/S)**: The distinction between *intuition* and *sensing* is established through the method by which an individual acquires information.
- **(T/F)**: The manner in which an individual makes decisions is determined by the contrast between *thinking* and *feeling*.
- **(P/J)**: The way in which individuals make lifestyle choices is differentiated by the dichotomy of *perceiving* and *judging*.

These categories were used in the Keirsey Temperament Sorter categorization and will be discussed in the following sections of the paper [15].

2.2 Hard skills

The technical competencies involved in software development are commonly known as the hard skills of a software developer. These skills can be adapted to various development domains depending on specific requirements.

Based on a questionnaire provided in [12], in general, 5 hard skills were identified in the average software developer.

- **Programming skills**: The ability to construct complex commands using programming languages at an advanced level
- **Spoken and written language skills**: Proficiency in several spoken and written languages for effective communication and collaboration on international projects.
- **Database skills**: Competence in database management for the purpose of designing and creating systems.
- **Field-specific skills**: Possessing expertise in a particular subject matter to assess and enhance technical work, while also providing guidance and instruction to others.
- **Scrum skills**: Having prior familiarity with the roles, practices, processes, procedures, and artifacts of the Scrum methodology.

3 What is the relevance of a great software team?

In this section, we discuss the relevance and advantages of well-formed software development teams in companies. The answer to the aforementioned research question will provide insight into the motivation of software team formation.
It is apparent that the adoption of a well-organized software development team's approach to production is more likely to lead to the successful completion of a project. In addition to it, there are several other factors that are benefited from such teams.

3.1 Employee performance

According to [9] software developers tend to lose their high performance and motivation when their potential is not fully executed. This is caused by the feeling of under-appreciation within the work environment. This leads to worse efficiency in individuals. Forming a well-structured software team ensures that all its members are assigned to do the job according to their capabilities. This policy not only leads to an improvement in the overall job satisfaction and productivity of employees but also offers cost benefits by minimizing unused resources.

In addition to the lost potential of an employee, according to [16], the quality of software development outcomes can also be severely compromised by poorly structured teams. Subsequently, such issues can result in reduced maintainability of the software product, thereby increasing the costs of the projects in the long run.

3.2 Team performance

An effectively structured software development team can boost the knowledge and skills of its members through effective communication among them. [2] Collaboration can help achieve this and result in the creation of highly qualified software products. A team with such an organized structure may greatly assist in the professional development of each member while also improving the performance of the team as a whole.

3.3 Team reputation

A software team that is well-organized and efficient can have multiple positive outcomes for a company. These outcomes include the ability to produce high-quality software products, which can ultimately lead to a good reputation for the team within the organization [7]. In addition to providing team members with a sense of pride and accomplishment, successful projects can also serve as motivation for them to continue doing a good job.

3.4 Company reputation

An effectively structured software team within a company can provide benefits beyond producing high-quality software products. It can attract other skilled professionals with interest in such a working environment [5].This can lead to a positive work culture and reputation, which can entice even more qualified candidates. Overall, more suitable employees have chances to be hired with an increase in

the number of candidates for software jobs.

4 What factors influence software team formation?

In order to facilitate the process of forming a software development team, it is relevant to understand what factors influence the quality of a software development team. If some of these factors are discovered and understood, they can be used to contribute to possible solutions for improving the formation itself. These factors are often attributes of the components of a software team, i.e. the individuals that comprise a team.

4.1 Attributes considered during formation

According to [8] 5 main types of attributes were always taken into consideration while forming software development teams. These attributes are the following. Technical attributes, they encompass abilities an individual possesses regarding expertise in the work itself. Individual costs, they account for the expenses individuals bring with them, in order to not exceed the budget. Personality traits, they regard personal characteristics of individuals. Interpersonal skills, are about the degree individuals have skills to communicate with each other. Social skills, they encompasses the quality of interactions an individual has with others. As becomes apparent, technical skills are by far the most considered attribute. Soft skills seem to get less attention, so researching the effect of more socially related attributes could be beneficial. According to [20], the following factors are to be considered when software teams are formed. Skills (soft), are characteristics belonging to individuals that encompass both technical and non-technical skills. Personality, traits that an individual possesses that dictate how they function. Team collaboration, is how a team interacts with each other, both professionally and informally. Project features, and constraints of a project. Other general factors encompass many smaller factors such as job satisfaction, age.

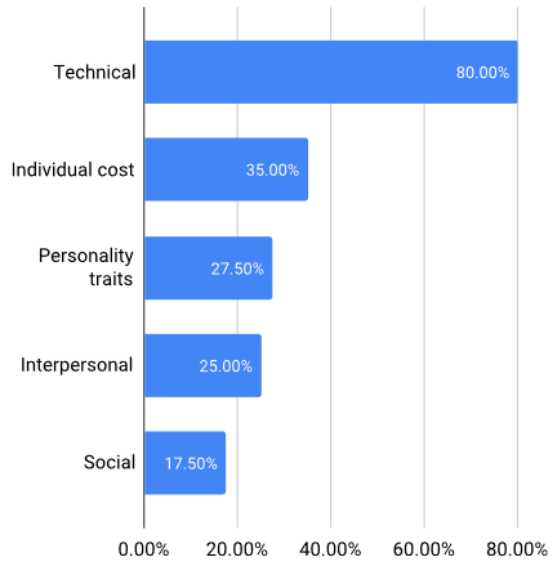


Fig. 1. Percentage of attribute types used during the team formation process, taken from [8]

4.2 Personality types

Personality types are part of the soft skills that are attributed to individuals and say something about their character. These personality types say something about how individuals make judgments, process information or gain energy from interaction with the outside world. These personality types consist of a combination of perceiving/judging (P/J), intuition/sensing (I/S), thinking/feeling (T/F), and extroversion/introversion (E/I). If teams consisting of individuals with varying personality types produce different results, this could indicate that certain sets of combinations of personality types result in more successful projects. Of course, the level of success of a project is not set in stone. To determine the effect of certain combinations of individuals and the effect on a project's success, one paper [14] made a questionnaire and interviewed both the project manager and individuals of a project team in two different projects. The questionnaire aims to determine the level of success of a project, how individual participants experienced the process of the project, and the individual personality types.

The paper did not provide the complete questionnaire, so only a sample of questions available on the questionnaire was published. The first section of the questionnaire contains questions meant for the project manager. These include questions regarding the project's: planned and actual effort, planned and actual size, planned and actual duration, planned and actual budget, success with respect to customer requests, etc. The second part of the questionnaire included questions meant for the project team and included questions regarding their role, time spent reworking, teamwork, learning, and self-improvement. The final part of the questionnaire determined what personality type the individual belonged to according to the Keirsey [15] temperament sorter. The questions in the first two parts of the questionnaire are used for an evaluation form that determines a project successful or unsuccessful. The evaluation form evaluates a project's success in terms of scope, effort, schedule, and budget. The project's social success and quality are also determined. Using certain criteria, all questions from the first two parts are categorized as either successful or unsuccessful. Using this, successful and unsuccessful projects were selected.

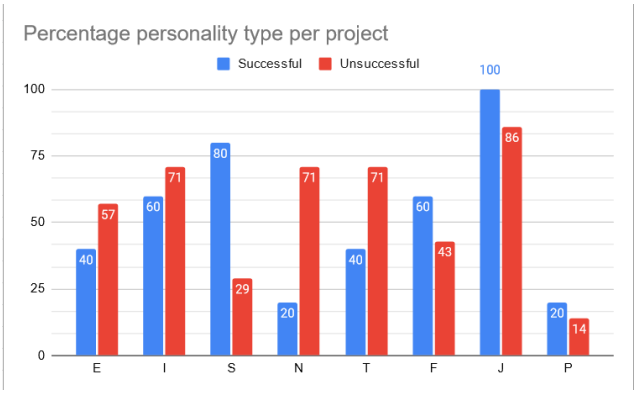


Fig. 2. Percentage personality traits for successful and unsuccessful project

The results of the percentage personality types present in both projects can be seen in figure 2. It is noteworthy that the judgment personality trait is dominantly present in both projects. The most apparent difference between the two teams is the percentage of sensing versus intuition. The successful project has a high percentage of sensing personality traits and little intuition, which is the opposite of the other project. These findings are not enough to draw a conclusion, but it could indicate that the sensing personality trait can have a negative effect on a software development team.

4.3 Social ties

The level of social connections between team members has an effect on the formation of software development teams. The paper [11] researches the effect of social ties on the formation of open-source software development teams. The findings of this study can be of value to research regarding the formation of software development teams in general because open-source projects often fail due to difficulty in ac-

quiring developers. This can be attributed to the fact that open-source projects do not provide rewards or payments to contributors besides learning and experience. Due to this fact though, factors that facilitate team formation in a setting where rewards are not part of the motivation for individuals will most certainly also be factors that apply in settings where rewards are part of the motivation. The paper proposes three hypotheses. The hypothesis is tested using data gathered from open-source project repositories hosted on SourceForge.net. At the time of publishing of the paper, this site contains hosting to over 100,000 projects and 1,100,000 subscribers. 1030 projects were randomly selected to be used in the research. To determine the level of social ties of the developers, data for the following criteria were collected per project:

- At least one developer joined the project within 1 month.
- Project initiator has preexisting social ties.
- Number of direct social ties initiator has prior to this project.
- Number of projects participated by the initiator.
- Number of project initiators.
- level of ambiguity in the project description.

Hypothesis 1 is: Projects whose initiators have preexisting social ties with the network are more likely to have other developers join the development team than those whose initiators do not have ties. This hypothesis was supported by the data gathered in the paper. Hypothesis 2 is: For those projects whose initiators have preexisting social ties with the network, the amount of such ties is positively associated with the probability of having other developers join the project team. This hypothesis was also supported by the data gathered in the paper. Hypothesis 3 is: For those projects whose initiators do not have preexisting social ties in the network, the experience of initiators is positively associated with the probability of having other developers join the project team. According to the data collected in the research, this hypothesis was not supported.
Judging by this paper, preexisting social ties are important for software development team formation in terms of the willingness of other developers to join, while experience plays less of an important role in terms of willingness to join.

5 What is done to improve software team formation?

After understanding the relevance of a great software development team, and some factors that influence the formation of software teams, it becomes relevant to dive into solutions for facilitating the software team formation process. Methods that improve the quality of formations increase the amount in which project management constraints can be satisfied. Since team formation can result in many combinations consisting of hundreds of potential members each with varying attributes and skills, doing this process manually can be time-consuming and lead to error. A solution or proposed method to automate or facilitate this process is therefore highly desirable. The better the method, the more optimal the team formation.

5.1 Characteristics of methods

A look at the mapping study [8] shares insight into some of the characteristics of proposed solutions in the research field of software team formation. The solutions found were all aimed at different objectives. 67.5% aimed at maximizing a project's requirements, while 7.5% focused on minimizing project costs, 7.5% focused on minimizing delivery time and 27.5% focused on improving relationships. This indicates that the methods overall attempted to make a project as successful as possible in terms of checking boxes with regard to its deliverables. The mapping study had 5 different categories of proposed methods, which consist of a total of 30 different methods. The individual methods are unfortunately not discussed in detail, which would have been beneficial in answering this research question. As can be seen in figure 3, most methods consist of search and optimization algorithms, including genetic algorithms, dynamic programming, etc. Then as can be seen in figure 4, most of the methods' goal for the output was multiple teams. Therefore, it is safe to assume that to facilitate software team formations, methods usually try to maximize project requirements by using search and optimization algorithms that result in multiple teams as outputs.

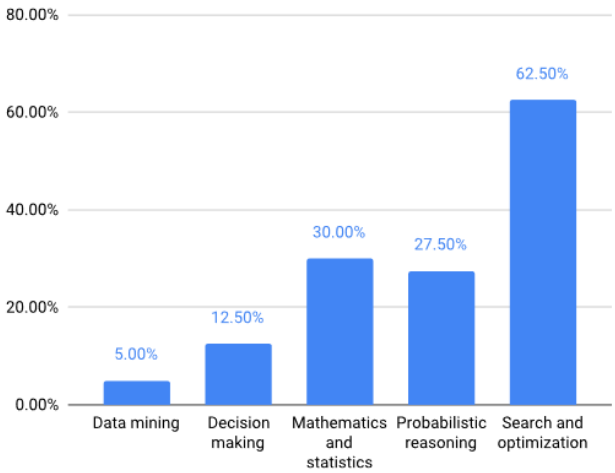


Fig. 3. Percentage of method categories, taken from [8]

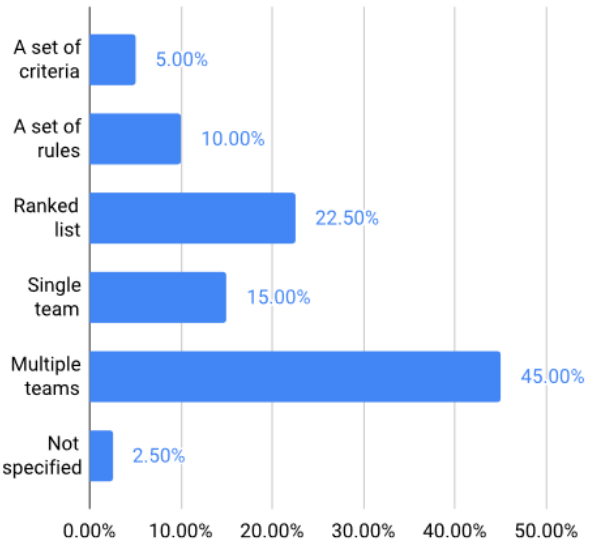


Fig. 4. Percentage of method outputs, taken from [8]

5.2 Methods for automation

The literature research [20] mentions numerous methods for automating the software formation process. The methods are unfortunately not described in detail. Some of the methods that are mentioned in this paper will now be shortly presented.

In the paper [10], a fuzzy algorithmic approach was proposed. It provided a model based on skills of individual developers to learn from each other. The algorithm uses size 2 tuples with linguistic terms and selects appropriate individuals based on their own skill set and those skills that were required for a given project. Their approach uses an aggregation algorithm that links individuals with skills similar to those

needed in a project. The skill relations are tuples, to show how two skills can influence each other, promoting learning.

Then [18] proposes a methodology known as Best-Fitted Resource. The approach is systematic and determines the level of compatibility and suitability for a skill set (belonging to a group comprised of individuals) and the skills that are required for a given project. This model effectively assigns individuals/skill sets most suitable for a job, making the most optimal selection. It takes attributes of candidates into consideration such as required skills, levels of expertise, and relative priorities of required skills for tasks. The problem with these two methods is that they only consider technical skills though, and as stated in RQ2, soft skills also contribute to the successful formation of teams.

Another study [17] used Ant Colony Optimization and event-based scheduling to allocate employees with a most suited task. It considers attributes such as skill set, salary, hours allowed to work, availability, etc. An event-based scheduler that enables modeling for the preemption of tasks and resource conflict. Then the ant-colony algorithm attempts to make an optimal solution with given human resources.

Another study [19] uses genetic algorithms to schedule resources according to priority and cost minimization. It incorporated practical matters into the fitness function. The fitness function is composed of a weighted combination of four fitness scores that take into account cost reduction, concentration efficiency, continuity considerations, and allocation balance. The results showed that when all four fitness scores are taken into consideration, the algorithm produces a more practical allocation of human resources, resulting in reduced multitasking time, fewer tasks assigned without considering task precedence, and more evenly distributed allocations, compared to an algorithm that solely focuses on minimizing the time span.

Other methods proposed in [4] go about dividing projects between suitable teams. It discusses a practical example in which a multi-criteria model was created to aid a global software company in allocating work for its distributed team. The model was developed in collaboration with software development project managers, using decision conferencing and multi-attribute value analysis. The model considers not only software engineering factors, but also ”soft” and strategic issues such as team satisfaction and training opportunities. In figure 5 can be seen that out of all papers, genetic algorithms seemed to be most popular, with other bio-inspired algorithms in second place, and fuzzy logic in third.

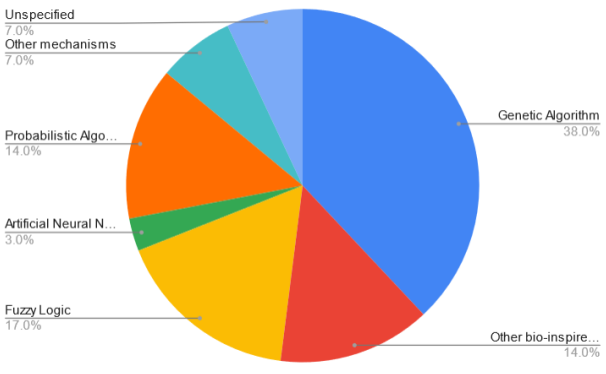


Fig. 5. Types of method, data from [20]

6 Discussion and Future Works

This section will now discuss some of the results found by answering the research questions by summarizing the papers used for this research and give a more concrete answer to the overarching question of what makes a great software team.

The first research question in this paper is answered by considering the idea of how important a well-formed team can be. We have looked into different aspects of software development teams and how they are benefited through such structure. It is apparent that such teams offer numerous advantages, each of which contributes in some manner to the successful completion of projects. Increased employee performance is one of the consequences. It was understood that the fulfillment of the employees' potential results in an increased sense of value in the team. Therefore producing more quality work. An additional advantage was discussed to be found in the enhancement of team performance and reputation. Effective communication among team members fosters the sharing of knowledge, which contributes to the improvement of individual and collective accomplishments and values. Finally, we've demonstrated how a strong software infrastructure is very helpful in attracting talented developers, opening up a variety of opportunities for improving a software team's abilities.

Research question 2 was about what factors influence software team formation. It became apparent that many attributes can be taken into account when forming a project (software) team. However, according to a mapping study [8], many found solutions take mostly technical attributes into consideration. This creates room for solutions that involve other attributes such as soft skills since it has become apparent that these do indeed influence the performance of a software team and the quality of their work, and the degree of success of a project they work on. The mapping does show that some methods do take soft skills into account, such as personality traits, and interpersonal and social skills. According to [14] the personality traits of individuals that make up a software team might affect the degree of success of a project. They found that the judgment personality trait is dominant in software teams for both successful and unsuccessful projects, and found that the sensing personality trait was present in successful projects while the intuition personality trait was lacking. This suggests that the sensing personality trait influences the results of software projects. Then social ties are also taken into consideration by paper [11]. This paper claims that developers are more likely to join a project if initiators have preexisting social ties before the start of a project and that the amount of ties is positively associated with the probability of developers joining. It also suggested that the skills of initiators without preexisting social ties do not influence the probability of other developers joining. This suggests the importance of social ties to influence the willingness of other developers to join a project.

Research question 3 is about what is done to improve software team formation. First, we consider the mapping study [8] to look at characteristics of methods that are used for software team formation. It became apparent that most of the methods had the objective of maximizing a project's requirement. The paper also makes clear that most algorithms are of the search and optimization type. Finally, it was clear that the most used output for the methods was an output that consisted of multiple teams. According to [20], some methods were discussed such as genetic algorithms, fuzzy algorithms, and more. The biggest downside of most methods was that they mostly considered hard skills and maybe some metadata about individuals such as their age, hourly wage, etc.

By answering these 3 questions, it becomes apparent that for future work there is a need for a software team methodology that puts an emphasis on soft skills. Most methods focus too much on hard skills and while delivering acceptable results, there is room for improvement. By researching and summarizing literature and papers, the impact of soft skills on team productivity becomes clear. An algorithm that includes not only hard skills but personality types and also social ties would be an interesting subject to research and could benefit project managers by formatting software teams more optimally. With regards to personality types, the algorithm looks at individuals with judging and sensing personality traits. The algorithm also looks for developers with preexisting social ties in a developing network.

7 Conclusion

In this research the question of what makes a great software team was considered. To answer this question, three sub-questions were used. These questions talked about the relevance of a great software team, what factors influence software team formation, and what was done to improve team formation. It was then found out that a great software team was relevant for project success, employee performance, team performance, team reputation, and company reputation. The factors that contributed to team formation were mostly technical skills, but the contribution of soft skills in the formation process seems promising. To be more precise, personality traits and social ties seem to affect team formation and performance. Finally, to improve software team formation, automation methods were used. These methods had as their objective to maximize project requirements, used search and optimization techniques and produced multiple team outputs. Genetic algorithms were mostly used. It can be concluded that a great software team is made of individuals with a suitable hard technical skill set appropriate for a specific job, while also possessing certain soft skill attributes such as certain personality types and social ties.

Acknowledgements

The authors wish to thank the following people. Rein Smedinga, Michael Biehl, and Renée Lutke for organizing the course which made the creation of this paper possible. Ayushi Rastogi for being the expert reviewer for this research paper. Fellow students involved in the reviewing process of this paper.

References

[1] F. Ahmed, L. F. Capretz, and P. Campbell. Evaluating the demand for soft skills in software development. *It Professional*, 14(1):44–49, 2012.

[2] J. M. Assbeihat. The impact of collaboration among members on team's performance. *Management and Administrative Sciences Review*, 5(5):248–259, 2016.

[3] A. Baratta. The triple constraint: a triple illusion. *PMI® Global Congress 2006—North America, Seattle, WA*, 2006.

[4] A. Barcus and G. Montibeller. Supporting the allocation of software development work in distributed teams with multi-criteria decision analysis. *Omega*, 36(3):464–475, June 2008.

[5] A. Barr and S. Tessler. How will the software talent shortage end? *American Programmer*, 11:2–7, 1998.

[6] K. Briggs. Myers-briggs typenindikator, mbti-manual (dt. bearbeitung von r. bents und r. blank). *Beltz Test GmbH 19952, Weinheim*, 1991.

[7] Y. M. Chen and C.-W. Wei. Multiagent approach to solve project team work allocation problems. *International Journal of Production Research*, 47(13):3453–3470, 2009.

[8] A. Costa, F. Ramos, M. Perkusich, E. Dantas, E. Dilorenzo, F. Chagas, A. Meireles, D. Albuquerque, L. Silva, H. Almeida, et al. Team formation in software engineering: a systematic mapping study. *Ieee Access*, 8:145687–145712, 2020.

[9] H. A. Eiselt and V. Marianov. Employee positioning and workload allocation. *Computers & operations research*, 35(2):513–524, 2008.

[10] V. Gerogiannis, E. Rapti, K. Anthony, and P. Fitsilis. A fuzzy linguistic approach for human resource evaluation and selection in software projects. *IEOM 2015 - 5th International Conference on Industrial Engineering and Operations Management, Proceeding*, 04 2015.

[11] J. Hahn, J. Y. Moon, and C. Zhang. Impact of social ties on open source project team formation. In *Open Source Systems: IFIP Working Group 2.13 Foundation on Open Source Software, June 8–10, 2006, Como, Italy 2*, pages 307–317. Springer, 2006.

[12] A. Hidayati, E. K. Budiardjo, and B. Purwandari. Hard and soft skills for scrum global software development teams. In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, pages 110–114, 2020.

[13] C. Jung and R. Hull. Psychological types (a revised ed.). *London: Routlege*, 1991.

[14] Ç. M. Karapıçak and O. Demirörs. A case study on the need to consider personality types for software team formation. In *Software Process Improvement and Capability Determination: 13th International Conference, SPICE 2013, Bremen, Germany, June 4-6, 2013. Proceedings 13*, pages 120–129. Springer, 2013.

[15] D. Keirsey and M. Bates. Please understand me 2: Prometheus nemesis book company. *Prometheus Nemesis Book Company*, 1998.

[16] M. Lavallée and P. N. Robillard. Why good developers write bad code: An observational case study of the impacts of organizational factors on software quality. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 677–687. IEEE, 2015.

[17] D. Monica. Scheduling and resource allocation for employees in software projects. *International Journal of Advanced Computational Engineering and Networking*, 2, 2014.

[18] L. D. Otero, G. Centeno, A. J. Ruiz-Torres, and C. E. Otero. A systematic approach for resource allocation in software projects. *Computers Industrial Engineering*, 56(4):1333–1339, 2009.

[19] J. Park, D. Seo, G. Hong, D. Shin, J. Hwa, and D.-H. Bae. Practical human resource allocation in software projects using genetic algorithm. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2014, 05 2014.

[20] W. Prashandi and A. Kirupananda. Automation of team formation in software development projects in an enterprise: What needs to improve? In *2019 International conference on advanced computing and applications (ACOMP)*, pages 16–22. IEEE, 2019.

[21] A. Roan and G. Whitehouse. Women, information technology and 'waves of optimism': Australian evidence on 'mixed-skill'jobs. *New Technology, Work and Employment*, 22(1):21–33, 2007.

Decentralized Federated Learning - Solutions based on Gossip Protocol and Blockchain

Nikhita Prabhakar and Shrushti Kaul

Abstract— Federated learning (FL) can securely train machine learning models on huge, rich, and private data. In FL, machine learning models are trained on local data sources and then merged to create a global model. However, unlike previously thought, FL is not always reliable to protect the privacy of the training data; there are instances wherein it's very possible to obtain private training data from publicly shared gradients [17].
The goal of this paper is to look into decentralized ways to implement FL through the usage of Gossip Protocol and Blockchain. Existing works on both, in the form of Gossip Learning [13] and Blockchain-based FL [12, 14] are analyzed in the paper and it is shown how they are a step up from vanilla centralized FL mechanism. However, while the implementation of Gossip Learning itself is based on many assumptions [2], the inclusion of blockchain makes the federated learning mechanism vulnerable to a whole new set of attacks.
Another goal of this paper is to study how a combination of both Gossip and Blockchain can be realized in FL. We give an informal guideline on how the said technologies can be clubbed together and also discuss implementations [16, 10, 15, 7] which use both technologies in the FL schemes, and compare certain security aspects which come into play in the said implementations.

Index Terms—Decentralized Machine Learning, Federated Learning, Gossip Learning, Blockchain in ML, Privacy-Preserving Machine Learning

1 Introduction

With the help of federated learning (FL), also termed as federated ML, joint learning, or alliance learning, [14] several devices can work together to jointly develop a model without sharing their data with a centralized server. Google first put up this idea in 2016 to address the issue of end users' local updates for Android mobile phone models [6, 14]. This approach to decentralized machine learning is especially helpful in industries like healthcare or finance where data privacy is an issue.
Federated Learning follows the "bringing the code to the data, instead of the data to the code" approach [1]. Each participant/client device trains a machine-learning model. Instead of sending raw data to a central server, The participants only submit the computed gradients (partial derivatives) of the local model parameters to the central server. The central server aggregates the gradients from all the participants to build a global model. This process is repeated until the global model converges or reaches a predefined stopping criterion. FL is well-suited in instances wherein data available from the clients in the federation is more insightful than the data that exists on servers, is privacy-sensitive or otherwise impractical to transmit to and store in servers, for instance, on-device item ranking, content suggestions for on-device keyboards, next word prediction [1].

The whole premise of FL or distributed machine learning is based on the fact that sharing model updates would not reveal the training data, and thus privacy is maintained. However, major studies, one of them being [17] by Zhu et al, have demonstrated that there is in fact information hidden in gradients. While it has been shown that gradients reveal properties of training data (e.g. property and membership inference), [17] presents a more challenging scenario of completely stealing local data in client nodes from publicly shared model gradients. They have achieved so through an optimization algorithm, termed "Deep Leakage of Gradients"(DLG) that generates dummy input and target labels and converts it to training input and target labels with the help of the public gradients in just a few iterations. DLG performed exceptionally well in generating training data from image classification and masked language model gradients.

Thus, newer implementations of federated learning are pivotal to mitigating the aforementioned privacy violations. One proposition made in recent research to achieve this is to eliminate the central component for performing the model-weights aggregation, thus giving birth to the domain of decentralized FL. This also helps in reducing risks of single-point-of-failure and possible man-in-the-middle attacks [14]. Two major technologies in the current research that are bound to be advantageous in decentralizing the vanilla FL architecture is peer-to-peer networks and blockchain. Peer-to-peer training architecture involves nodes exchanging their locally computed gradients within themselves, thus discarding the need for an aggregation server. A well-known example is "Gossip Learning", in which participants mimic the gossip protocol to relay gradient updates. On the other hand, blockchain-based infrastructures offer an intuitive solution for FL implementation. There have been several proposals to implement FL with blockchain technologies, as they provide benefits such as verification of computation performed by local nodes during local training. Thus, these two techniques of implementing FL in a distributed way are seen as a better alternative to centralized FL. While both technologies serve their purpose in their own right, they do come along with their own set of weaknesses, which could perhaps be mitigated when combined.

The main objectives of this paper are as follows:

- Analyze shortcomings with recent works in Gossip Learning and Blockchain-based FL.
- Deduce how Gossip Protocol and Blockchain could be combined to provide a decentralized FL framework.
- Analyze current works that are trying to club both technologies as well.

The paper proceeds as follows: in section 2 (Background), we introduce the concept of federated learning, followed by going in-depth about decentralized FL methods using gossip protocol and blockchain aka Gossip Learning and Blockchain-based FL and see how incorporating these techniques into the vanilla FL mechanism proves advantageous. In section 3 (Pitfalls in Gossip Learning and Blockchain-based FL), we discuss the potential disadvantages that exist with current Gossip Learning and Blockchain-based FL methods, in section 4 (FL Solutions based on both Gossip Protocol and Blockchain), we infer how the said technologies can be combined to manifest decentralized FL and have a discourse on recent research that strives to achieve

- *Nikhita Prabhakar, n.prabhakar@student.rug.nl.*
- *Shrushti Kaul, s.u.kaul@student.rug.nl.*

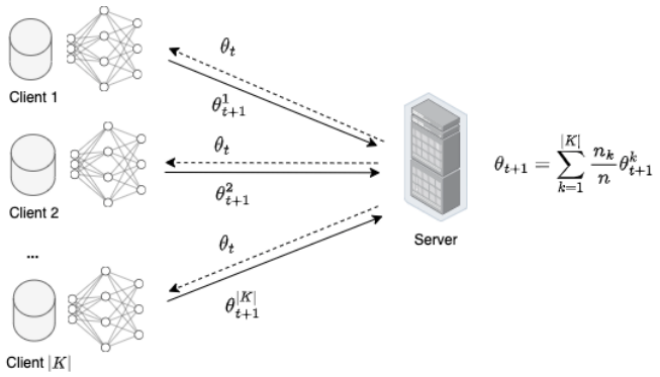


Fig. 1. Federated Averaging [1]

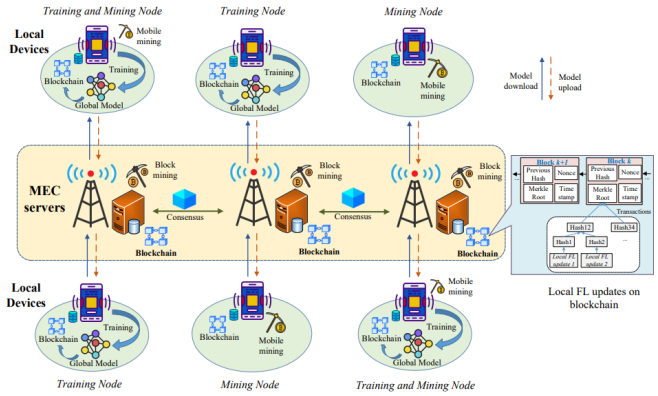


Fig. 2. Blockchain-based FL paradigm for FLchain from [12]

the same while keeping certain metrics in mind, which are also discussed in the same section, and finally finish with a section for the conclusion and future work.

2 Background

In this section, we shall shed some light upon the vanilla, centralized federated learning mechanism, which is followed by talking about Gossip Learning and Blockchain-based FL and how due to their decentralization capabilities, they triumph as better FL frameworks than the originally proposed one.

2.1 Federated Learning (FL)

The aim of federated learning is to perform machine learning on decentralized data. This also enables clients/edge devices to perform machine learning with privacy by default. An example of a classic FL strategy for model optimization is via Federated Averaging (FedAvg) [9]. FedAvg tries to minimize an overall global loss, which is essentially a weighted average of the individual clients' losses. Each of the losses computed on the local sites is weighted by the size of a client's data set. A round in the algorithm is depicted in Figure 1. The algorithm is executed for model training for a number of rounds 't'. A fraction of the K clients is sampled and the current round's weights are sent to the selected fraction of clients. The clients in turn run gradient descent for a certain number of epochs and send the updated weights back to the server. Once the server has received all the weights, it then performs the weighted average of the client model's updates.

2.2 Gossip Learning as an alternative to FL

Gossip Learning was first introduced in [13] as a way to exploit peer-to-peer technology's ability to achieve scalability in a cheap fashion, but also due to its "potential for privacy-preserving solutions". This implementation involves models that perform a random walk in a network of peers, which uses every visited peer's local dataset for weight-updation. Algorithm 1 is run on all peers in the network. The algorithm consists of a periodic activity loop and a procedure for

Algorithm 1 Gossip Learning Scheme in Ormandi et al. [13]

```
initModel()
loop
    wait(Δ)    p ← selectPeer()
    send modelCache.freshest()to p
end loop
procedure ONRECEIVEMODEL(m)
    modelCache.add(createModel(m, lastModel))
    lastModel ← m    end procedure
```

dealing with approaching models. Every newly generated model is cached, and the model that has been cached the longest is replaced by newer ones. The most recent model is assigned to a random peer, which is taken off in [13] via a "gossip-based implementation of peer-sampling". Later, Hegedus et al. [4] present this as a replacement for FL, quoting the fully decentralized approach of the gossip-learning scheme. This acts as a benefit also because it eliminates the possibility of single-point-of-failure. [4] considered performance as a metric for comparison, in terms of convergence time and quality of the model and used logistic regression as the main machine learning model. Results of the paper show that gossip learning outperformed federated learning in two out of the three datasets, which is counter-intuitive considering the fact that gossip learning has a slower aggregation than federated learning, due to its fully distributed nature. Thus, Hegedus et al. strongly encourage the use of fully decentralized algorithms in the area of multi-node machine learning.

2.3 Blockchain-based FL

Nakamoto initially introduced blockchain as the fundamental ledger of the renowned Bitcoin cryptocurrency [11]. In essence, a blockchain refers to a replicated and distributed append-only database that allows a network's participants to maintain a tamper-resistant sequence of data. The main idea behind the notion of incorporating blockchain in FL is that data does not belong to a central entity. In this paper, we stick to the work of Nyugen et al. [12] as an illustration of blockchain-enabled FL, which introduces a generic FL-blockchain architecture, namely "FLchain", which entails prominent features of blockchain-FL system designs. In simple terms, a blockchain network operates by participants disseminating their data, and certain nodes, known as miners or validators, collect and store the received data in blocks. Through a decentralized consensus mechanism, the network selects a leader miner for a sequence of epochs. The epoch leader broadcasts their block to the network, and other nodes store it in their local memory, with each block maintaining a hash link to the previous block. A generalized blockchain-based FL architecture is depicted in Figure 2.

This architecture specifically caters to the use case of mobile edge computing (aka MEC), wherein MEC servers are considered as a point for data collection in the scenario of edge computing. FLchain transforms intelligent MEC networks into decentralized systems. Here, the MEC servers are in charge of performing blockchain mining while mobile devices can be employed to either execute local training or mining operations or carry out both in case they possess the capability. The general steps towards performing blockchain-based FL via FLchain are as follows:

- The MEC servers are assigned a learning objective with their related devices. MEC servers allocate their computational resources to run the blockchain consensus (or mining). The associated devices to the MEC servers perform the FL training.
- Every training device initializes a local model and trains it with its local data and then sends the local model to the associated MEC server via blockchain through a transaction

- MEC servers aggregate all transactions sent by clients and creates the block, which is followed by the servers participating in the mining process for the verification of the newly generated block and to reach consensus among all MEC servers. The verified block is appended to the blockchain and broadcasted to all training devices.

- Local devices retrieve all the local updates from other devices by downloading the block, thus permitting them to compute the "global" model directly based on various aggregation rules, for instance, weighted average(FedAvg [9]). The training process is executed until the global loss function converges or desired accuracy is achieved.

Blockchain replaces the requirement of having a central component, which mitigates significant communication costs while also providing a high level of security to perform FL training via ledgers that are immutable [12]. Moreover, FL can take advantage of rather innate properties of blockchain to alleviate some of its major identified drawbacks, like centralized processing, lack of incentive for clients in the federation and low robustness [14].

3 Pitfalls in Gossip Learning and Blockchain-based FL

While the inclusion of distributed technologies lists a plethora of benefits, they, unfortunately, bring forth their own set of shortcomings in the implementation and execution of FL. This section discusses these shortcomings both for the case of Gossip Learning and Blockchain-based FL.

3.1 Issues with Gossip Learning

A study by Giaretta et al. [2] states that the implementation of Gossip Learning in the original literature [13] is highly dependent on certain assumptions, which would not be upheld in uncontrolled environments. These assumptions are: a) each device stores a single data point, also referred to as the fully-distributed data model, b) each device can communicate with every other device in the network, and c) all devices possess similar communication and processing speeds. In order to determine the conditions under which the protocol continues to function and those under which it breaks, the researchers simulated the protocol under various workloads while raising these assumptions to various degrees.
While lifting the first assumption regarding a "fully distributed data model" doesn't bring forth an undesirable execution of Gossip Learning, removing the latter assumptions had the opposite effect. The following subsections shall throw some light on this.

3.1.1 Presence of restricted network topologies

A set of experiments were performed in [2] to assess the impact of limited communication networks on the performance of the protocol, results of which are displayed in Figure 3. The findings for these experiments demonstrated that well-connected topologies (ex: community-based graphs, Barabasi-Albert) depict convergence speeds in an order similar to that of a fully-connected network. It can be concluded that gossip learning is only efficient on topologies where there is a combination of low distance and high redundancy, which is indicative of good expansion. It is evident that low link redundancy topologies, for instance, trees and rings, clearly show a much slower convergence.

3.1.2 Uneven communication and processing capabilities

Another set of experiments in [2] were done to test gossip learning for instances wherein nodes had differing speeds. Assigning speeds to these nodes randomly, even from an expansive range, leads to the same convergence in the same number of epochs, acting as if all the speeds were of equal measure. A clear indication of the same is depicted in the first and second graphs in Figure 4.

However, there is a drastic change observed in the model's behaviour when the speed and data distributions are correlated. The last graph in Figure 4 depicts a fluctuation in the functioning of gossip

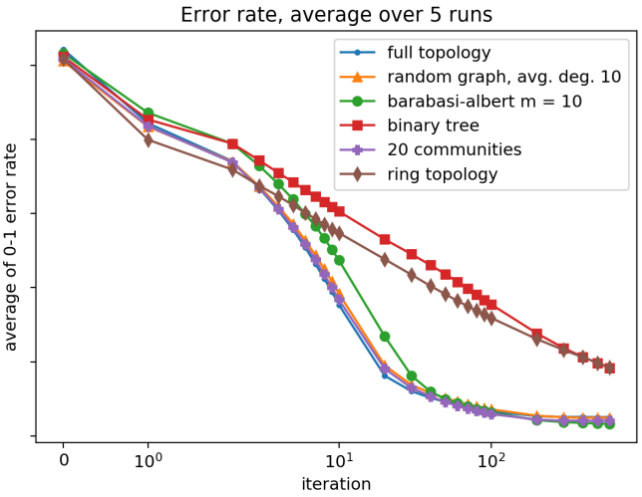


Fig. 3. Comparison of different network topologies in [2] to observe speed of convergence in Gossip Learning

learning while training on a cosine dataset. The fluctuation is realized by storing all data points with $x > 0$ (shown as red) on nodes that are much faster than all nodes storing data points with $x < 0$ (shown as blue). The model in this case rapidly favours the red data points and thereby converges to a negative slope.

Thus, gossip learning is not adept at handling topologies where its characteristics correlate with the features of the dataset.

3.2 Issues with Blockchain-based FL

Despite novel architectures of using FL with blockchain (for instance, FLChain [12]) exhibiting great potential, there are issues of various kinds that may not be taken care of in their implementation.

3.2.1 Security Challenges

As described in [12], blockchain still possesses a set of own security issues, some of which include: a) forking attacks, which can easily derail the training process for the nodes participating on the blockchain, b) fake parameter updation from an adversary portraying an honest client.

3.2.2 Heterogeneity and Communication Challenges

Just like in Gossip Learning, even communications that take place in FL training are easily affected by unbalanced and non-IID data. Also, a major increase in traffic congestion can be observed with an increasing blockchain network, contributing to a slow or even erroneous FL training rate of convergence. While compression methods have been recommended to make transferring model updates to and from client faster, they could increase variance in the model updates and make it harder to reach the correct convergence result.

3.2.3 Plagiarism Challenges

The FLchain system has another flaw wherein a "lazy" node could save up more of its resources for mining than training by simply copying model updates of a different client, thereby having a much higher probability of earning rewards. This might push the honest client from participating in the FL training and degrade the performance of the overall FL model.

4 FL Solutions based on both Gossip Protocol and Blockchain

One of the paper's objectives is to elucidate how Gossip Protocol and Blockchain could be fused together in a way that they complement each other for implementing decentralized FL. While introducing fully decentralized machine learning, work in [5] indicates that it can be

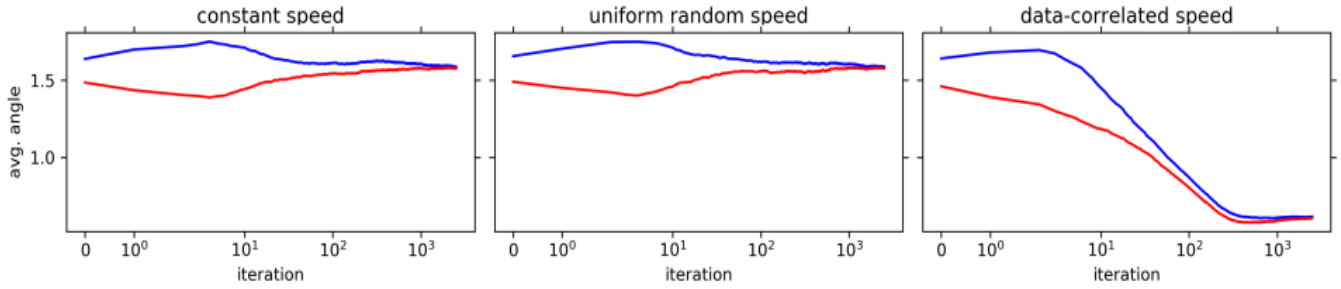


Fig. 4. Behaviour of gossip learning with various speed distributions displayed in [2]

practically realized with the help of a distributed ledger, which can help take over tasks like global model aggregation and participation of a federation consisting of different companies and organization with the help of smart contracts. This section is split into two sections, wherein the former section shall present an informal mechanism of how both techniques can go hand in hand. The latter section shall discuss recent research that has taken place where an FL implementation tries to incorporate both Gossip Protocol and Blockchain.

4.1 Informal mechanism of merging Gossip Protocol and Blockchain in FL

Below-mentioned is a mixed approach towards distributed machine learning in which, staying true to their purpose, blockchain manages the updation and enforces privacy, whereas the communication of the updates in the model is done via the gossip protocol. A take on the same could look like the following:

1. All the clients or peers in the network makes use of the gossip protocol to share their gradients with one another. Every client encrypts its local updates and shares them with the blockchain.

2. The blockchain aggregates the updates using a consensus mechanism, after which the ”global” model is stored on the blockchain, which can be copied by the clients to perform further training.

4.2 Analysis and discussion of current works combining both technologies

This section goes into the research available that exploits both blockchain and gossip protocol for an FL solution.
While performing research, there were very limited sources that tried merging the implementation of a gossip learning framework with blockchain. The one promising lead we found in this regard was the work of Giaretta et al. [3], wherein the federated learning implementation translated to using gossip learning on top of the Ethereum blockchain. But, even this source was just introducing the architecture and did not really include any implementation or experimentation of any sort with the explained federated learning model.
Thus, the works we are going to analyse are more so decentralized FL techniques which are essentially built on top of blockchain and heavily rely on gossip protocol to broadcast and communicate local model updates between the clients of the federation. There are four papers that we shall discuss and compare in the following sections.

4.2.1 Current Works

A. PiRATE [16]
In 5G networks, every mobile device is able to partake in distributed learning. Thus, the availability issue of distributed machine learning becomes irrelevant. Nevertheless, the safety of the model is compromised as the distributed learning system is now vulnerable to Byzantine attacks while updating model parameters and aggregating gradients amongst multiple learners. PiRATE is introduced as a secure computing framework based on the blockchain sharding technique to ensure byzantine resilience for distributed learning by providing security of gradient aggregation and model parameters.
In this algorithm, each node would possess a randomly generated identifier. Committees composed of c members shall be formed after the node is allowed to participate by a centralized permission/access control component. All participants of the committee are aware of the identities of trustworthy nodes with a high probability. During the training phase, some nodes will be swapped out for trustworthy nodes in order to impede the progress of a slowly-evolving adversary. Once part of a committee, the members of the committee communicate their local gradients using gossip protocol which then can be aggregated by the leader, into a block. This block can be validated by the committee and a consensus via blockchain after a certain time period. This partial aggregation result is transferred to the neighbour committee. Once every committee reaches a consensus, they communicate locally agreed data with their neighbour committees, eventually, a globally agreed aggregation value is obtained by all the nodes.

B. BEAS [10]
BEAS is a blockchain-based FL system that ensures training data privacy using gradient pruning to prevent the poisoning of training data and model gradients. It claims to be one of the first blockchain-based FL systems whose accuracy is comparable to that of centralized frameworks while having a linear growth of computation and communication overheads with respect to the number of participants in the federation. An overview of the BEAS paradigm can be seen in Figure 5. As seen from the paradigm, BEAS uses Fabric blockchain. The BEAS framework scheme is as follows: Clients of the federation create anonymous identities with the help of a Membership Service Provider (MSP). Any client can create a new channel, define the training parameters and train on their private data to produce a genesis block, which is appended to the channel ledger. The remaining clients seek the latest block in the ledger, which is used to initialize a model, which is trained using local data to create new local gradients. The client sends these local gradients to the endorsing peer for generating a new block and sharing it with the Ordering service. This service provides consensus on the act of ordering blocks and appends them to the ledger of every endorsing peer's ledger using the gossip protocol. The endorsing peer triggers a 'Merge' chain code of the incoming blocks once they cross a threshold. This chain code aggregates the blocks using federated averaging to generate a new global block, which is committed to the channel ledger. The client keeps repeating these steps until desired convergence is achieved.

C. DeepChain [15]
While Deep Learning(DL) requires a humongous amount of data and training to achieve the required accuracy, The usage of FL in deep learning can help improve the accuracy of the model by training data in different nodes, in the same amount of time. The local gradients are then communicated which is then used to generate the complete model. This however exposes the deep

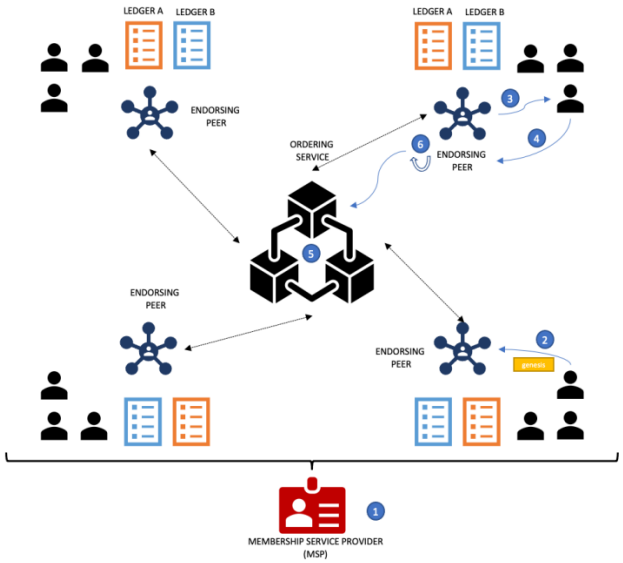


Fig. 5. Blockchain-based FL paradigm for BEAS [10]

learning model to a variety of security issues, mainly revolving around the communication of gradients and parameters across nodes. DeepChain, a value-based incentive mechanism, guarantees the privacy of data for each node and enforces the correct behaviour of the participants with the help of blockchain. Cryptographic techniques are used to ensure data confidentiality and auditability. In DeepChain, we have every single node(party) as part of a blockchain. These parties have the data which is to be trained locally. Those parties involved in the training of the same DL model form a cooperative group. The parties of this group receive incentives or penalties based on their contributions. Once local model training is completed by the party, trading of the local gradients is done, which essentially attaches the gradient values to smart contracts, generating a transaction. A committee then verifies and arrives at a consensus on the addition of the new block. The security while trading the gradients is ensured by a cryptographic algorithm which provides encryption of up to 80 bits and to ensure a unique cipher is generated for a party, the parameters are converted into a vector for the same. This addition of a new block is then communicated to the other neighbours using a gossip protocol. Consequently, we obtain a fully trained model.

D. BLADE-FL [7]
BLADE-FL suggests blockchain-assisted autonomous federated learning. Here, each client broadcasts the trained model to other clients, aggregates its own model with received ones, and contests to create a block before its local training of the next round. BLADE-FL's learning ability and global loss function upper limit are assessed. This is followed by checking whether this bound is convex with regard to the number of overall aggregation rounds K and optimising computing resource allocation to minimize the upper limit. In BLADE-FL, every client acts as the trainer as well as a miner. It is assumed that all the clients in the distributed system have the same computing power. Initially, the node trains the local model and is added as part of the transaction. This transaction is communicated to all other clients with the help of a gossip protocol. The client, now acting as a miner, receives all the local models which would subsequently be used to generate the global model. And then, the block comprising the local model is validated. Once it's validated and accepted by most of the miners, it's added to the local ledgers. The verified block is now used to update the local models, enabling all the clients to have the global model. Since each of the clients takes part in mining and training, the single-point failure of the centralized system is mitigated while maintaining the privacy of the Federated FL system with the help of blockchain.

4.2.2 Analysis of Current Works

The above-mentioned works are a few novel and innovative approaches used to combine blockchain with gossip protocol in an FL implementation which will help in mitigating security and privacy issues. We could see four major aspects being considered in the manifestation of these papers, indicated in Table 1. We shall briefly discuss these four features and how the selected works, if they are able to, provide support for the respective features.

	Byzantine Attacks	Scalability	Data Poisoning	Model Poisoning
PiRATE	No	Yes	Yes	No
BEAS	Yes	Yes	Yes	Yes
DeepChain	Yes	Yes	Yes	Yes
BLADE-FL	No	No	Yes	Yes

Table 1: Indication of support for certain aspects in recent decentralized FL methods

A brief description of factors we think should be kept in mind while creating a decentralized FL solution is as follows.

- **Byzantine Attacks** In the context of FL, when a model is being trained on a local node, on top of the local data, one can not guarantee that the local training process itself is correct or the node is i) actually malicious in nature and trying to perform training incorrectly, or ii) an attacker has taken control of a previously honest node and made it malicious over time. Both of these cases, eventually lead to a compromised distributed FL framework.

- **Scalability** Scalability has also been considered in this discussion since a lack thereof could lead to lossy updates of the model updates that shall eventually lead to an erroneous model convergence and thereby an incorrect distributed machine learning model which won't behave as expected.

- **Data and Model Poisoning** A specific way in which byzantine attacks can be realized is through data and model poisoning attacks. Work in [8] defines a data poisoning attack that happens during the local data collection phase and a model poisoning attack that happens during the local model training process. Essentially, the goal of the said attacks is to modify the behaviour of the target model in some undesirable way.

Implementation of checks against any forthcoming byzantine attacks is taken care of either via centralized access control, like in PiRATE, that checks credit scores and is able to tackle a slow adversary attack or through the inclusion of a penalty scheme, like in DeepChain, which checks the results of local functions executed by the peers and rewards/penalizes the peer based on verification of the respective local results. The penalty scheme in DeepChain also provides assurance with model poisoning since the training behaviour of the nodes is being assessed.
Protection against the combination of byzantine and data poisoning attacks can be provided with the inclusion of byzantine tolerant gradient aggregation algorithms like MultiKRUM, like in PiRATE and BEAS. Data poisoning attacks can also be tackled with the inclusion of adding Gaussian noise to the data.
An efficient way in which model poisoning is being tackled is via gradient pruning, an efficient strategy which is also recommended in the reference regarding the earlier mentioned DLG attack in [17]. We have made an inference that byzantine failures and the poisoning attacks go hand in hand, but just implementing a poisoning-attack proof FL implementation, will not guarantee an FL implementation resistant to byzantine attacks as seen for BLADE-FL. Thus, there are

other factors as mentioned in [10] which need to be considered other than poisoning attacks which can help in the development of much more secure implementations of decentralized FL, for instance, the addition of more encryption policies, asynchronous updation, and so on.
Coming to scalability, works, especially PiRATE and DeepChain perform a great job with regard to scalability by having smaller groups of committees that validate model updates within themselves and then share these local updates across the different committees to reach a global consensus, allowing for a lot of amortized executions. respectively. BLADE-FL on the order hand takes advantage of its Hyperledger Fabric-dependent implementation and exploits multi-channel blockchain architectures for training different FL models simultaneously.

5 Conclusion

This paper introduces federated learning and discusses the inclusion of decentralization technologies in FL like gossip protocol and blockchain, in efforts to mitigate security concerns that arise with the implementation of federated learning with a central aggregation component. It is observed that while the inclusion of said technologies promises decentralization and avoids single-point failure, the existing works around the same still harness weaknesses that need to be addressed. This paper also presents an informal guideline on how one can utilize both Gossip protocol and Blockchain for the implementation of distributed FL in Section 4.1. A study of the latest selection of papers shows that tackling byzantine and poisoning attacks is one of the primary goals in recent decentralized FL methods that incorporate both blockchain and gossip protocol in their implementation. It can be observed that a lot of issues in distributed FL come down to ensuring the integrity of the data or gradients which are being shared and maintaining the data present in the node. But the field of decentralized FL is still in its infancy and just providing support for poisoning attacks will not suffice for a completely secure distributed FL implementation.

6 Future Work

More research could be performed in the following directions:

- First, as realized from the analysis of recent works, most of the works on FL that try to include both gossip protocol and blockchain in their FL solution tend to confront only a subset of desirable features for decentralized FL. Future work could include coming up with solutions that try to address the majority of issues that come along with implementing FL in a distributed fashion.

- Second, it was deduced that there is no concrete research found on clubbing blockchain specifically with gossip learning to manifest decentralized FL. Putting theory to practice and actually implementing/experimenting with FL architectures including gossip learning and blockchain, like one mentioned in [3] can help bridge the gap and give a different perspective on how merging gossip protocol and blockchain can elevate decentralized FL.

Acknowledgements

The authors would like to thank expert reviewers Prof. F. Turkmen and A. R. Ghavamipour for their invaluable feedback. They would also like to extend their gratitude to L. Pulles, G. Kilinkaridis and S. Somasundaram for their constructive feedback towards the paper.

References

[1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design, 2019.

[2] Lodovico Giaretta and Šarūnas Girdzijauskas. Gossip learning: Off the beaten path. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1117–1124, 2019.

[3] Lodovico Giaretta, Ioannis Savvidis, Thomas Marchioro, Šarūnas Girdzijauskas, George Pallis, Marios D. Dikaiakos, and Evangelos Markatos. Pds2: A user-centered decentralized marketplace for privacy preserving data processing. In *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*, pages 92–99, 2021.

[4] István Hegedüs, Gábor Danner, and Márk Jelasity. Gossip learning as a decentralized alternative to federated learning. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, 2019.

[5] Peter Kairouz, Brendan Avent Aurélien Bellet Mehdi Bennis Bhagoji Arjun Nitin ... McMahan, H. Brendan, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2):1–210, June 2021. Publisher Copyright: © 2021 Georg Thieme Verlag. All rights reserved.

[6] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[7] Jun Li, Yumeng Shao, Kang Wei, Ming Ding, Chuan Ma, Long Shi, Zhu Han, and H. Vincent Poor. Blockchain assisted decentralized federated learning (blade-fl): Performance analysis and resource allocation, 2021.

[8] Lingjuan Lyu, Han Yu, Jun Zhao, and Qiang Yang. *Threats to Federated Learning*, pages 3–16. 11 2020.

[9] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. 2016.

[10] Arup Mondal, Harpreet Virk, and Debayan Gupta. BEAS: blockchain enabled asynchronous & secure federated machine learning. *CoRR*, abs/2202.02817, 2022.

[11] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. May 2009.

[12] Dinh C. Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N. Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, and H. Vincent Poor. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal*, 8(16):12806–12825, 2021.

[13] Róbert Ormándi, István Hegedűs, and Márk Jelasity. Gossip learning with linear models on fully distributed data. *Concurrency and Computation: Practice and Experience*, 25(4):556–571, may 2012.

[14] Youyang Qu, Md Palash Uddin, Chenquan Gan, Yong Xiang, Longxiang Gao, and John Yearwood. Blockchain-enabled federated learning: A survey. *ACM Comput. Surv.*, 55(4), Nov 2022.

[15] Jiasi Weng, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. Cryptology ePrint Archive, Paper 2018/679, 2018. https://eprint.iacr.org/2018/679.

[16] Sicong Zhou, Huawei Huang, Wuhui Chen, Zibin Zheng, and Song Guo. Pirate: A blockchain-based secure framework of distributed machine learning in 5g networks, 2019.

[17] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients, 2019.

Representation of Women on Stack Overflow: A ten-year overview on participation, challenges, and research

Joep Scheltens, and Davide Rigoni

Abstract—The low participation of women in STEM-related subjects has been the topic of discussion in many research in the past, especially if we consider the field of engineering which has a very low number of women participating. This low participation becomes even more concerning if we inspect the participation of women in the popular QA website for programmers, Stack Overflow in which women's participation is even lower than in other computer science departments, such as in big tech companies and open source projects.

In this paper, we present a ten-year overview of the representation of women on the website Stack Overflow by focusing on how the research on the topic has evolved during the last ten years and what are the common challenges, found in the state-of-the-art, experienced by women in Stack Overflow that prevent them from interacting and using the website as much as men. This, along with an inspection of the evolution of the percentage of active women on the website from 2012 to 2022, will give a comprehensive overview of how the topic changed over the years.

We found that the representation of women in Stack Overflow has not increased in the last ten years although the research on the topic has increased lately, with more studies being performed from 2016 onwards. Nonetheless, the main challenges women experience, as found in the literature seemed to remain the same throughout the years which indicated in Stack Overflow an inamicable environment for women to participate in. This gives a bright outlook on the future of the topic where hopefully more studies built on top of already existing findings will propose more solutions to solve the topic.

Nevertheless, we suggest a couple of possible improvements to the website, as found in the literature, that could help women engage more in the platform, such as the possibility of adding social feedback functionality.

Index Terms—Stack Overflow, Women representation, Gender diversity

1 Introduction

In the field of the computer science industry, a very important resource is the ability to interact with the developers' community to ask and answer technical questions as well as share knowledge. This is an important learning tool for programmers who in return can learn programming concepts and solve certain tasks or bugs by just interacting with the community. For this purpose, the application Stack Overflow was launched in 2008 as a valuable resource for asking and answering technical questions in the field of computer science. However, as different studies show [21] [16] the platform's user base is predominantly male, with the percentage of participating females being consistently less than 10%. This data is particularly worrisome especially if we compare those numbers to other data from the computer science sector. For example, the percentages of women in big informatics companies such as Microsoft and Google stand at around 20% [18], and women constitute around 18% of the total graduates in computer science in the USA [2].

This shows a general imbalance between the women working and studying in the field of computing science and the ones using the website Stack Overflow. This low participation of women in Stack Overflow has already been documented in a study of 2012 by Vasilescu et al. [21] in which it shows that men were more represented than women on the website and were participating more and earning more reputation.

The purpose of our study is to investigate the changes that happened from 2012, the year of publication of the study [21], until now to explore the current state of research on the topic through the case study of the paper by Vasilescu et al. [21] and how new studies on the topic have emerged since then, especially focusing on studies whose main goal is the identification the common challenges and limitations experienced by women on Stack Overflow. Hence in this paper, we will answer the following research questions:

1. *RQ1: Has the participation of women in Stack Overflow changed between 2012 and 2022?*
2. *RQ2 What are the common challenges and limitations women encounter in Stack Overflow?*
3. *RQ3: How are empirical findings on women's representation in Stack Overflow utilized in subsequent research studies?*

While RQ2 and RQ3 focus mostly on how the research on the topic evolved since 2012, the latter concerning the original paper by Vasilescu et al. [21] and the former about the identification of common challenges and limitations women experience in Stack Overflow, the main goal of RQ1 is to find evidence on why this topic is still as relevant now as it was in 2012.

In section 2 of this paper, we will discuss in more detail the current state of the art regarding women's participation in computer science, section 3 will focus on the methodology used to perform the research for the paper. Section 4 will consist of the results found from the literature review on the topic and on the participation of females in Stack Overflow while in section 5 we will discuss those results. Finally, in section 6 we will go over the threats to the validity of our study while sections 7 and 8 will present the conclusion of the paper and possible future works.

2 Background and related work

Women have historically been underrepresented in the field of Computer Science. Research has found that females only comprise of 15% of CS mayors in Ireland, less than 20% in the United Kingdom, and 14% in Australia [23]. Some researchers think that the myth of women not being interested in computer science and engineering is a self-fulfilling prophecy. For example, studies have shown that activities marked with a gender stereotype make the opposite gender less interested in the activity [14].

Women's representation and contributions to computer science are important for several reasons. For example, in the workplace diversity has a positive impact on key organizational performance aspects

- *Davide Rigoni is with University of Groningen, E-mail: d.rigoni@student.rug.nl.*
- *Joep Scheltens is with University of Groningen, E-mail: j.scheltens.1@student.rug.nl.*

like recruiting talent, strengthening customer orientation, increasing employee satisfaction, improving decision-making, and enhancing a company's image [8]. Not enough diversity in the workplace might lead to missing out on a wider range of perspectives and ideas, which in the world of computer science, could potentially lead to the development of better products, systems, and technologies that address the needs of a diverse range of users.

Since Computing Science is not very gender diverse, the userbase of platforms like Stack Overflow being predominantly as well male makes sense.. The 2012 [21] study revealed that around 7% of website users in 2012 were women, meaning that women in the Computing Science community are less likely to use the website than men in the Computing Science community. Since this study, more research has been done about the representation of women on Stack Overflow. Some researchers have found that feminine or mostly feminine users have the lowest reputation score on average, which is the central measure of success on the platform [3]. And research has also found that women are even more discouraged from contributing to open-source projects. Surveys on Stack Overflow have shown that comparing non-open-source settings to open-source settings, men are twice as likely to contribute to open-source projects than women [24]. The challenges women face that make them feel discouraged from participating in Stack Overflow is, therefore, a concern, so these will be discussed more in detail in section 4.2.

Other research includes analysis of the yearly surveys that Stack Overflow provides [18] [4]. However, research that analyses the Stack Overflow survey data does not restrict itself to gender diversity. Research has also been done for finding methods to make Stack Overflow more gender-inclusive [11]. Research on other platforms, like GitHub, has also been performed. For example, the original paper by Vasilescu et al. [21] was followed by a study about datasets for social diversity studies of GitHub teams [22]. This study found that women are underrepresented on GitHub as well, meaning that the underrepresentation of women on Stack Overflow is not necessarily platform exclusive. What separates our research from related work is that it focuses specifically on changes to gender diversity on Stack Overflow since the 2012 study.

3 Methodology

In this section, we will discuss the main steps and methods we employed to do this research, focusing on the research query and paper selection performed for RQ2 and the classification performed for RQ3.

3.1 Paper Selection

To gather literature on the topic we performed a short literature review on Google Scholar using the following research query:

```
  ("women") AND ("challenges" OR
"representation") AND ("Stack Overflow")
```

Through this query, we obtained literature describing the challenges women encounter on the website Stack Overflow which is necessary to answer RQ2 as well as get contextual data on women's representation in Stack Overflow that we used throughout the paper.

Furthermore, we decided to perform a second literature search and shortly investigate other environments related to women's representation in fields related to computer science. For this reason, we added the keywords "GitHub", "STEM" and "Open Source" which allowed us to gather data on women's representation in the fields of Open Source projects as well as STEM subjects which consist of any academic subject falling under the disciplines of Science, Technology, Engineering, and Mathematics. We wanted to gather some data on the representation of women in those fields and, for this research, were not particularly interested in the challenges hence our second research query was:

```
  ("women") AND ("representation") AND
("GitHub" OR "STEM" OR "Open Source")
```

It is important to note that the only exclusion criteria for the literature found in our study were whether the study was published after 2012 or not. For this research, we did not consider the ranking of papers and the journal they were published in since we feared that we might not have enough literature to answer RQ2; this shortcoming is discussed more in-depth in section 6.

3.2 Classification

In our study, RQ3 refers to the use of empirical findings on the topic of women's representation in Stack Overflow and how are those findings used for future research. To develop an answer to this research question we took the paper "Gender, Representation and Online Participation: A Quantitative Study of Stack Overflow" by Vasilescu et. al. [21] which is one of the first studies published on the topic.

The study in question develops a tool named 'gendercomputer' which can identify the gender of users based on their usernames and other attributes. This tool is used to gather data from Stack Overflow users in 2012 to see the participation and the level of engagement of women on the website in comparison to men.

We gather the papers that cited this paper from 2012 to 2022 and analyzed each citation individually and divided them into three different groups:

- *Black Box citations*: those citations which only mention the study as a source that has performed a study on Stack Overflow and gender differences but do not mention any of the results obtained. Furthermore, cases in which the citation is relative only to the 'gendercomputer' tool also constitute a case of black box citation.

- *White Box citations*: those citations which use one or more of the findings of the paper in their study.

- *Unidentified citations*: this category groups together citations that were either wrong, from papers that got removed, or not accessible to us.

We proceeded to classify all citations based on the aforementioned categories as shown in the Result section.

3.3 Kappa coefficient

Since the classification of the citations is performed by two researchers, we have to make sure that although both researchers analyze a different set of citations the resulting categories will be the same. Therefore we have to calculate the level of agreement between the two of us since it's a great indication of whether we understood the categorization correctly and if we'd also agree in some citations in which the use of the data would not be clear.

In order to measure this level of agreement we used the Cohen Kappa coefficient, which provides a formula to measure the overall agreement between two researchers in the classification of different items [9]. The value of Kappa is calculated as follows:

$$k = \frac{p_0 - p_c}{1 - p_c} \tag{1}$$

The values p_0 and p_c in equation 1 correspond respectively to the percent agreement and the chance agreement which is the proportion of agreement expected by chance [20].

The highest possible Kappa value is 1 which signifies complete agreement while the lowest is 0 which signifies complete disagreement.

In our study, we both performed the classification on 40 items separately and then calculated the Kappa coefficient which in our case ended up being 0.81 which shows a high level of agreement.

4 Results

In this section, we will show the results obtained for each one of the research questions along with a short explanation of them.

4.1 Results: RQ1

In Figure 1 we can see the percentage of female users in Stack Overflow from 2012 to 2022. This data comes from the official annual surveys of Stack Overflow except for the data from 2012 which was taken from the study conducted by Vasilescu et. al. [21]. As explained more in detail in Section 6, this is one of the main threats to the validity of our study, since the data from 2012 comes from the use of the 'gendercomputer' tool to infer the gender of the user of Stack Overflow which inevitably results into a more global analysis than the annual surveys in Stack Overflow.

Furthermore, it's important to point out that there is no data on female participation in the platform for 2013 and 2014 due to the fact that the survey of Stack Overflow started including questions regarding gender from 2015 onwards. Additionally, we were not able to find clear data on female participation on the website in the literature for those two years so we decided to leave them out.

Even without the data from 2013 and 2014 the graph in Figure 1 gives a clear overview of the trend of female participation in the website Stack Overflow from 2012 to 2022.

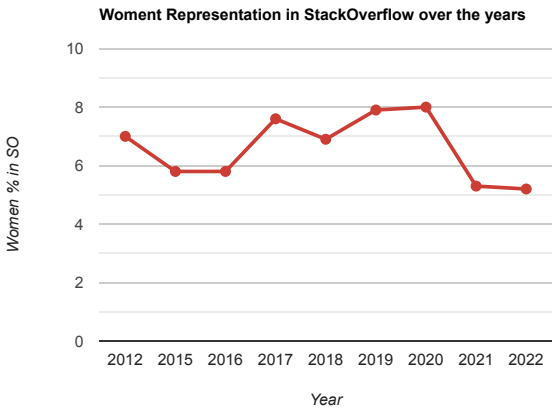


Fig. 1. Women representation in Stack Overflow from 2012 to 2022 as found in the survey of Stack Overflow apart from 2012 in which the results were obtained from the study by Vasilescu et. al. [21] are used.

4.2 Results: RQ2

For RQ2 we wanted to investigate the common challenges that women experience on the website Stack Overflow which deter them from using it and participating in its community. The result consists of findings found in the literature review which are referred to as the common reasons why women tend to engage and participate less than men in Stack Overflow.

It's important to mention that the list of challenges that will be discussed is not necessarily for women only but can also be shared by men or gender-fluid individuals, but the reason they're presented here is that those challenges are found to be especially discouraging for women.

The common challenges found are:

- *Fear of negative feedback* received by the community is one of the main reasons women do not engage in Stack Overflow [7][10][12].
- *Less confidence in programming skills* than men which can indicate the reluctance to participate in discussions in Stack Overflow [15][19].
- *Unwelcoming and negative environment* for new users [7][15]. This point is also highlighted in the Survey of 2019 of Stack Overflow in which women express the need to make Stack Overflow a "friendlier" environment.
- *Use of inappropriate and masculine language* [7] as well as the fact that using feminine usernames proves to be a disadvantage [3]. This is also present in the Survey of 2020 of Stack Overflow in which women commonly refer to the "toxic" language as one of the main points to change.
- *Individualistic and competitive nature of the website* [5], especially the scoring structure which seems to disfavour women[3].
- *Lack of peer parity* is an interesting point, which consists of the lack of women interacting with each other on the platform and can be proven to be a reason why women do not engage in Stack Overflow discussions [6].

4.3 Results: RQ3

In table 1 we can see the final result of the classification of citations to the paper taken as our case study for the research question. We can see that the total amount of citations are almost evenly distributed between black and white box with also a small percentage of unidentified citations.

To better show the results of the classification process we are going to present examples of citations that were classified either as black or white box.

As an example of white box citation, the paper "Social Influence and learning pattern analysis: Case studies in Stack Overflow" by Paul et. al. [17] mentions that "The findings of B. Vasilescu et al. [reference] confirm that men constitute the vast majority of contributors to Stack Overflow". This is clearly a white box citation since it mentions the findings of the paper in question.

On the other hand, an example of black box citation is found in the paper "SOTorrent: reconstructing and analyzing the evolution of Stack Overflow posts" by Baltes et. al. [1] mentions that "Regarding the population of SO users, studies described properties such as gender [reference]". In this case, it's also clear the nature of the citation since it mentions that the study revolved around gender on Stack Overflow but does not mention any of the results.

Lastly, we also want to include an example of black box citation in which the 'gendercomputer', as explained in section 3.2, tool was cited; this is the case of the paper "An Analysis of Design Process and Performance in Distributed Data Science Teams" by Maier et. al. [13] mentions that "Further, existing tools can infer the gender of a GitHub user based on their username [reference]".

Table 1. Citation history of the paper by Vasilescu et. al. [21]

Total	Black	White	Unidentified
215	46% (99)	40% (87)	13% (29)

Additionally, we performed a year analysis of the citations, dividing them by the year the paper citing the study from Vasilescu et. al. [21] was published. The results of this analysis can be found in Figure 2.

On top of that, Figure 3 shows how those papers are distributed per year and whether they are white or black box citations.

5 Discussion

In this section, we are going to discuss and further analyze the results found in section 4.

5.1 Discussion: RQ1

As mentioned in section 2, the lack of women in Computing Science is concerning. Somewhat surprisingly, the participation of women on Stack Overflow seems to not have changed too much over the last ten years. Figure 1 provides a clear overview of female participation in the website from the last decade, and the data shows that the percentage of female users has remained relatively consistent, and even seemed to have dropped during this period. The lack of change is significant since it means that either not enough effort has been put into making

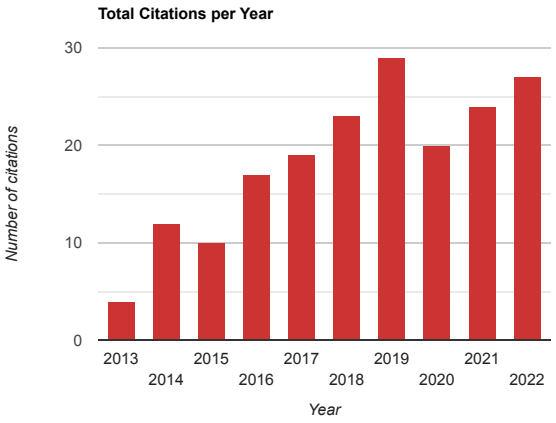


Fig. 2. Total Citation history of the paper by Vasilescu et. al. [21] from 2013 to 2022

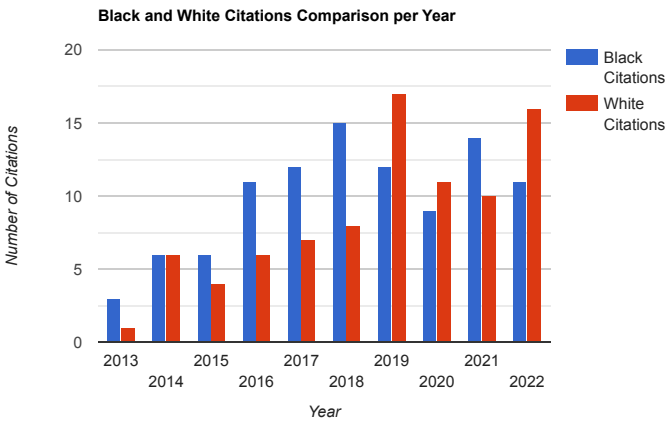


Fig. 3. 'Black and White Box Citation' history of the paper by Vasilescu et. al. [21] from 2013 to 2022

the website more gender inclusive, or the efforts that have been made did not work. This could be a topic for future research.

5.2 Discussion: RQ2

The findings for RQ2 revealed several challenges that are commonly faced by women on the platform. These challenges can be addressed in several ways to promote a more inclusive environment for women on Stack Overflow. One possible strategy is to develop policies and guidelines that promote respectful and inclusive language on the platform. This may involve being more strict on moderating, for example, rude or arrogant answers to questions.

Another approach could be to create opportunities for women to connect and support each other on the platform. This could involve creating forums or groups specifically for women to discuss programming topics or providing mentorship programs to connect female programmers with experienced mentors.

Stack Overflow could also consider changing its scoring structure to ensure that it does not disadvantage women. This may involve reevaluating the criteria for scoring or developing alternative scoring mechanisms that do not favor one gender over the other.

Finally, Stack Overflow could take steps to make the platform more welcoming to new users, including women. This could involve either providing more resources specifically for new users or adding labels to questions that are asked by newcomers. This might result in the people that answer the question being more lenient and forgiving on the asked questions.

5.3 Discussion: RQ3

The number of papers that cite the original papers means that there is a significant interest in the topic. Also, the amount of references to the original paper seems to have increased per year. The amount of white-box citations, meaning the citations that actually use the data of the original paper, has also increased per year. The years 2020 and 2021 are the clear exceptions to this pattern though, which could be due to the COVID-19 pandemic.

6 Threats to Validity

Some threats could be about the validity of the surveys on Stack Overflow. The Stack Overflow surveys rely on self-reported data from users, which may be subject to bias. Respondents may be hesitant to report negative experiences or may over-report positive experiences. Or the opposite, meaning that users that had a negative experience are more likely to want to give feedback. The surveys may also suffer from response bias if certain groups of users are more likely to participate than others. For example, if women are less likely to respond to the survey than men, then the survey results may not accurately reflect the experiences of women on the platform.

Also, the list of papers that cited the original paper [21] might miss some papers that cited it. Besides, no other metric has been used for the black box, white box, and year labeling than the number of papers in categorization. For example, It has not been analyzed whether or not black box citations were mainly in papers that were of lower or higher quality than those in white box citations. Or that in 2019 the number of citations was the highest, which does not necessarily mean that for this year this topic has been more well-researched. The amount of data is also possibly too little to discuss patterns.

Another threat would be that the authors are biased to the findings of the paper by Vasilescu et al. [21], since the supervisor of this current paper was also an author on that paper. The threat might be that the authors are more inclined to follow the same direction as that paper, instead of coming up with new ideas for research. To mitigate this threat, the authors have engaged with relevant literature outside of Vasilescu et al.'s study to provide a more comprehensive perspective on the issue of the under-representation of women in the tech industry.

Another potential threat to the validity of this paper is the quality of the referenced paper. The reliability of the data and the conclusions drawn in the current study rely heavily on the accuracy and credibility of the sources used. To mitigate this threat, we added the date to. However, due to the subject being niche, and the number of papers on

it being fairly low, the selection of papers was mainly done based on personal judgment of a paper's quality, instead of having predefined metrics for judging the quality. Therefore, the paper might become harder to reproduce.

Also, when we use data from the Stack Overflow surveys, or reference research that uses Stack Overflow survey data, we assume that this data is correct. However, the survey respondents are self-selected, which means that only those who are interested in participating or who feel strongly about the survey topic are likely to respond. This could introduce bias into the data, as those who respond may not be representative of the broader population of developers. Additionally, respondents may not always provide accurate or truthful information, either intentionally or unintentionally, which could further compromise the validity of the data. Finally, the questions themselves may be subject to interpretation or may not be comprehensive enough to capture the full range of opinions or experiences of the survey population.

Lastly, the graph in Figure 1 both contains data from the Stack Overflow surveys, as well as quantitative data gathered from the 'gendercomputer' [21]. Here it is assumed that the quantitative analysis as performed in the original study is comparable to the results of the survey. However, research has been done that states that the survey data and the qualitative analysis data are in line with one another [16].

7 Conclusion

The representation of women in the website Stack Overflow is, as shown in the Result section, a topic of research and study, with new papers being published and using previously found data on the topic. This new trend of research hopefully will be able to solve the common challenges women experience on the website that prevents them from participating in the website as much as men.

While the increase in research on the topic constitutes a positive outlook for the future, a point of concern is that representation of women in Stack Overflow has not increased from 2012 to 2022, with it even dropping below 6% in the last two year which shows how the challenges identified in the website are yet to be solved.

Nonetheless, as mentioned above, studies on how to overcome those challenges have already been done, identifying possible solutions to the problem. One such idea it's presented in the paper by Maftouni et. al. [12] in which the authors present the idea of a "Support" button that the users can utilize as social feedback as a way to reward prosocial behavior or negative one. This change was positively accepted especially by the women participating in the study who identified in that a way to make the Stack Overflow community a more positive and welcoming environment. Another study by May et. al instead proposes to create an alternative reward system [15] which consists of the equalization of points gained from upvotes on a question and on an answer since the current state of Stack Overflow rewards answers more than questions and women are found to ask in average more questions than men in the platform [15].

In conclusion, we can see how the research and understanding of the topic regarding women's representation in Stack Overflow have evolved and improved in the past 10 years, with more studies focusing on the topic and trying to find a solution to the under-representation of women on the website by trying to overcome the common challenges women face. Nonetheless, it seems like the percentage of women on the platform hasn't improved over the past 10 years which shows that the topic it's still very relevant and important for future research to focus on.

8 Future Works

Future research on the topic should mainly focus on how to solve the challenges identified in this paper as that could prove an important step forward to increase the representation of women on Stack Overflow. Another possible step for future researchers is to look toward the representation and participation of individuals who do not identify with the binary categorization of men and women.

Furthermore, we think that future studies should also focus on the representation of other minorities in the developer community and specifically on Stack Overflow. In fact, from the Stack Overflow surveys of 2021 and 2022 we can see how the strong majority of user identifies as heterosexual with less than 10% of users identifying as "Bisexual", "Gay or Lesbian", "Prefer to self-describe" and "Queer". And, another under-representation is found in the ethnicity of Stack Overflow users which according to the surveys of 2021 and 2022 consist of roughly 60% of white or European users. We think that it's important to investigate those topics further to make Stack Overflow and generally the developer community more inclusive in the future.

References

[1] S. Baltes, L. Dumani, C. Treude, and S. Diehl. Sotorrent: Reconstructing and analyzing the evolution of stack overflow posts. In *Proceedings of the 15th International Conference on Mining Software Repositories*, MSR '18, page 319–330, New York, NY, USA, 2018. Association for Computing Machinery.

[2] N. A. Bowman, C. Logel, J. LaCosse, L. Jarratt, E. A. Canning, K. T. Emerson, and M. C. Murphy. Gender representation and academic achievement among stem-interested students in college stem courses. *Journal of Research in Science Teaching*, 59:1876–1900, 12 2022.

[3] S. J. Brooke. Trouble in programmer's paradise: gender-biases in sharing and recognising technical knowledge on stack overflow. *Information Communication and Society*, 24:2091–2112, 2021.

[4] O. A. Dada, G. Obaido, I. T. Sanusi, K. Aruleba, and A. A. Yunusa. Hidden gold for it professionals, educators, and students: Insights from stack overflow survey. *IEEE Transactions on Computational Social Systems*, 2022.

[5] P. M. J. Dubois, M. Maftouni, and A. Bunt. Towards more gender-inclusive qas: Investigating perceptions of additional community presence information. *Proceedings of the ACM on Human-Computer Interaction*, 6, 11 2022.

[6] D. Ford, A. Harkins, and C. Parnin. Someone like me: How does peer parity influence participation of women on stack overflow? In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Oct. 2017.

[7] D. Ford, J. Smith, P. J. Guo, and C. Parnin. Paradise unplugged: Identifying barriers for female participation on stack overflow. volume 13-18-November-2016, pages 846–857. Association for Computing Machinery, 11 2016.

[8] V. Hunt, D. Layton, S. Prince, et al. Diversity matters. *McKinsey & Company*, 1(1):15–29, 2015.

[9] T. O. Kvalseth. Note on cohen's kappa. 1989.

[10] M. Maftouni. Gender consideration in the design of online knowledge-sharing qa platforms, 2022.

[11] M. Maftouni, P. Marcel, J. Dubois, and A. Bunt. "thank you for being nice": Investigating perspectives towards social feedback on stack overflow.

[12] M. Maftouni, P. Marcel, J. Dubois, and A. Bunt. "thank you for being nice": Investigating perspectives towards social feedback on stack overflow.

[13] T. Maier, J. F. Defranco, and C. Mccomb. An analysis of design process and performance in distributed data science teams.

[14] A. Master, A. N. Meltzoff, and S. Cheryan. Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. 118, 2021.

[15] A. May, J. Wachs, and A. Hannák. Gender differences in participation and reward on stack overflow. *Empirical Software Engineering*, 24:1997–2019, 8 2019.

[16] M. Nivala, A. Seredko, T. Osborne, and T. Hillman. Stack overflow – informal learning and the global expansion of professional development and opportunities in programming? pages 402–408, 2020.

[17] S. S. Paul, A. Tripathi, and R. R. Tewari. Social influence and learning pattern analysis: Case studies in stackoverflow. In S. K. Bhatia, K. K. Mishra, S. Tiwari, and V. K. Singh, editors, *Advances in Computer and Computational Sciences*, pages 111–121, Singapore, 2018. Springer Singapore.

[18] K. K. Silveira, S. Musse, I. Manssour, R. Vieira, and R. Prikladnicki. Reinforcing diversity company policies: Insights from stackoverflow developers survey. volume 2, pages 119–129. SciTePress, 2019.

[19] K. K. Silveira, S. Musse, I. H. Manssour, R. Vieira, and R. Prikladnicki. Confidence in programming skills: Gender insights from StackOverflow developers survey. In *2019 IEEE/ACM 41st International Conference*

on Software Engineering: Companion Proceedings (ICSE-Companion). IEEE, May 2019.

[20] S. Sun. Meta-analysis of cohen's kappa. *Health Services and Outcomes Research Methodology*, 11:145–163, 12 2011.

[21] B. Vasilescu, A. Capiluppi, and A. Serebrenik. Gender, representation and online participation: A quantitative study of stackoverflow. pages 332–338. IEEE Computer Society, 2012.

[22] B. Vasilescu, A. Serebrenik, and V. Filkov. A data set for social diversity studies of github teams. In *2015 IEEE/ACM 12th working conference on mining software repositories*, pages 514–517. IEEE, 2015.

[23] J. R. Warner, S. N. Baker, M. Haynes, M. Jacobson, N. Bibriescas, and Y. Yang. Gender, race, and economic status along the computing education pipeline: Examining disparities in course enrollment and wage earnings. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*, pages 61–72, 2022.

[24] P. Wurzelová, F. Palomba, and A. Bacchelli. Characterizing women (not) contributing to open-source. pages 5–8. Institute of Electrical and Electronics Engineers Inc., 5 2019.

Comparison of sampling methods in the validation of machine learning models

Christodoulous Hadjichristodoulou, and Herman (H.J.) Lassche

Abstract— In order to tune a machine learning algorithm and assess the predicted performance, an accurate performance estimation is essential. Due to this, a variety of performance estimators have been developed over the past few years, most of them based on cross-validation and bootstrap sampling. In this paper we provide an overview on a number of such techniques and examine their overall predictive performance by aggregating the results various research teams have reported on their experiments where they perform comparisons between them. We reach the conclusion that no single method is universally the best, but the suitability of each one depends on the conditions of the experiment.

Index Terms—Cross-validation, bootstrapping, model selection, performance estimation

1 Introduction

Despite the additional hurdles and computational costs it presents, model validation is a beneficial process for every project which aims to predict future outcomes based on past trends.It is not advisable to depend on the predictions of a model without verifying and validating it. In critical domains such as healthcare and autonomous vehicles, errors in object detection can result in severe casualties due to incorrect decisions made by the machine. Validating the machine learning model during the training and development stages can ensure that these kind of errors are minimized as much as possible. It can also help the researchers discover possible mistakes made during the development of the model or irregularities in the data used for training.

The data that researchers have to work with while developing a machine learning model is frequently limited. Researchers want to train the best possible model while also estimating the model's performance. Yet, given the limited data, it can be difficult to train the model on sufficient enough data and obtain a reliable model estimation for novel data. Over the years, a variety of sampling methods have been created in an effort to find an accurate performance prediction.

Multiple writers have compared and analyzed different approaches and variations in terms of the precision of the resulting performance evaluations, they give benefits and drawbacks of various sampling techniques in machine learning model validation. In this paper we provide an overview of a number of these approaches and aggregate the results of the authors' experiments.

As there is a wide variety of approaches, and that number keeps growing, we consequently felt the need to compile an overview of all the various techniques. Also, considering the benefits and drawbacks covered in this paper will help you choose a method.

We chose a few papers that cover established strategies and others that present novel ideas. We aimed to have some publications that discussed fundamental techniques and others that introduced more sophisticated strategies. We reach the verdict that no single technique can be declare as the objectively best for all cases. Currently, it seems like researchers prefer using variants of Cross-Validation, but Bootstrapping should be considered when dealing with irregular datasets, as we explain further below.

We will first provide some background information on performance estimators, cross-validation, and bootstrapping in section 2. We build on this information and examine particular approaches in more detail in section 3. Then, in section 4, we will contrast those strategies with one another and examine their performance. The paper will be concluded in Section 5.

- *Christodoulos Hadjichristodoulou is with University of Groningen; E-mail: c.hadjichristodoulou@student.rug.nl.*
- *Herman (H.J.) Lassche is with University of Groningen E-mail: h.j.lassche@student.rug.nl.*

2 Background

In this section we describe a few fundamental terms that are used throughout the whole paper: performance estimation, Cross-Validation and Bootstrapping.

2.1 Performance Estimation

Performance estimators are typically used for two different types of tasks. The first type is to choose the best fit of algorithms and hyper-parameter settings. Predicting performance on novel data is the second objective of the estimator [17]. We will focus on the second purpose in particular in our paper, namely predicting the performance on novel data. The estimation of a prediction-estimator can therefore be compared to the actual performance of the model.

We will frequently reference a study of Borra et al. [2]. In their study they use the extra-sample-error. This error, which is also known as the prediction error, describes the estimated function capacity to predict all potential values of the covariate variables X, for $f(X)$.

2.2 Cross-validation

Cross-validation (CV) [12, 8], also found in literature as K-fold cross-validation, is a method for performance estimation. With this method the training data is partitioned into K non-overlapping subsets, each one of which will act as a validation set for a model. After partitioning the data, we train a model on K-1 of these subsets and then estimate its performance on the remaining one. This process is repeated K times, once for each subset that is left out of the training set, and the final error estimate is calculated as the average of the K estimates. This approximation corresponds to a model that will have the same parameters as the ones used for the models trained on the K-1 subsets, but that will be trained on the whole training dataset. Figure 1 visualizes the procedure of splitting the dataset for the case where K=5. The dataset is split into 5 non-overlapping folds.

2.3 Bootstrap sampling

Bootstrap is widely used to determine performance estimation [1]. It is based on the theory that you can approximate the true distribution of data by sampling the true dataset repeatedly [7]. This resampled dataset often strongly matches the true distribution of the data.

In essence, we can build a new dataset that has the same distribution as the training/true data by randomly selecting datapoints from the original distribution, hence having roughly the same mean, variance, etc. For instance, the resampled dataset will have more instances of the datapoints that were frequent in the original dataset. Rare datapoints will likewise be uncommon in the resampled dataset.

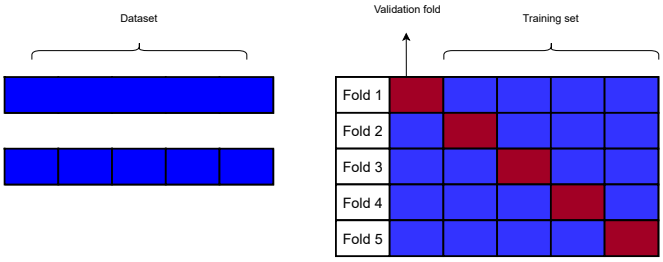


Fig. 1. Visualization of Cross-Validation with K=5.

Figure 2 serves as an illustration, showing how the sampled set produces roughly the same mean and distribution. It is possible to run statistical measure tests on this dataset because the sampled set and the original dataset share the same characteristics. The performance of the model when used with this dataset may be used as a representative of its performance with new data.

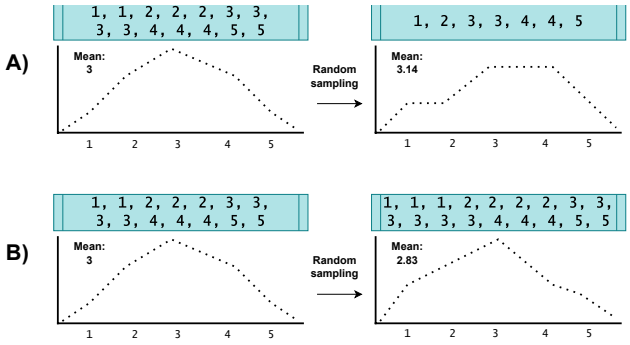


Fig. 2. An illustrative example of bootstrap sampling; A) Using a sampled dataset that is smaller than the original dataset. B) Using a sampled dataset that is larger than the original dataset (with replacement)

3 Estimaton approaches

In this section we provide an overview of various resampling techniques that are used currently to estimate the performance of a model.

Paper selection Our supervisor, Michael Biehl, introduced us to the subject of comparing cross-validation methods with bootstrap approaches. He sent us two papers that discussed adapted versions of the fundamental CV and bootstrap techniques. To find further papers, we looked through the RUG smartcat database. For this, we used queries consisting of a combination of the following words: `bootstrap, cross-validation, vs, comparison, performance estimation, evaluation`

Many papers came from this. We debated the relevance of these papers with our supervisor, Biehl. These were all relevant. To avoid overloading the paper, we decided to choose just three papers. The remaining found papers would be used to provide further context and details on the fundamental techniques. One of the three papers that were chosen discusses different bootstrap techniques, another discusses different cross-validation techniques and novel approaches that combines cross-validation with bootstrap, and the third discusses bias corrections. All this approaches will be discussed in the following section. We do the comparison based on the reported performance.

3.1 Cross-validation Sampling Methods

Below we summarize some variations of CV that improve on the basic method either by means of providing a better estimate or being computationally more efficient.

3.1.1 Cross-validation with Tuning

Cross-validation with Tuning (CVT) [17] is a method that employs cross-validation in order to tune the hyper-parameters of a model. The process involves employing cross-validation once for every set of hyper-parameters we want to test and then selecting the one with the lowest reported loss. The final model is then trained on the whole training dataset with the chosen hyper-parameters and returned.

3.1.2 Repeated Cross-validation

The method of cross-validation can be noisy and may yield different results with different splits of the data. A way to counter this variance is to repeat the process multiple times, each time with different partitioning of the data. This means that in a K-fold cross-validation repeated X times scheme, the basic cross-validation procedure will be performed N times, each with different splits and $K * X$ models will be trained in total. The final error estimation will be the average error across all folds and all repeats.

3.1.3 Leave-one-out Cross-validation

Leave-one-out (LOO) cross-validation is an extreme case of CV where the size of the subsets is exactly one. This means that N models are trained, where N is the number of data points, each one trained on all datapoints but one. This method obviously incurs a heavy computational overhead and should thus be avoided for large datasets.

3.1.4 Nested Cross-validation

Nested Cross-Validation (NCV) [18] is used for evaluating the performance of a predictive model in a way that is less prone to overfitting than traditional cross-validation. The idea behind nested cross-validation is to have an outer loop and an inner loop of cross-validation. The outer loop is used to split the data into training and testing sets, while the inner loop is used to perform model selection and hyperparameter tuning. In the outer loop the data is split into training and testing sets. The testing set is put aside and not used until the very end. For the inner loop the training data is split into training and validation sets. The model is trained on the training set and evaluated on the validation set. Different hyperparameters and model architectures can be tried to determine the best performing one. Once the best model and its corresponding hyperparameters are selected, it is trained on the entire training set from the outer loop. The trained model is then evaluated on the testing set. These steps are repeated several times with different splits of the data in the outer loop to get an estimate of the model's generalization performance.

3.1.5 The Tibshirani and Tibshirani Protocol

Tibshirani and Tibshirani [16] proposed a novel approach, known as the TT method, to address the computational burden of NCV by estimating and correcting for the bias of CV without requiring additional model training. This method considers each fold as an independent dataset and uses it to estimate the level of optimism in the process of selecting the best configuration from multiple options. The TT method calculates the bias by comparing the loss of the final, chosen configuration with the one selected in a given fold.

3.2 Bootstrapping Sampling Methods

Here we will provide an overview of Parametric and Non-Parametric bootstrap. The apparent error will serve as the starting point for both of the bootstrap methods being considered. This a very basic estimate [3]. It is defined by the average of the loss function of the training dataset. The apparent error use the same data for both model training and model evaluation. As a result, it is known for being too optimistic yet is also easy to compute.

3.2.1 Non-Parametric Bootstrap

The non-parametric bootstrap takes a number of n random samples from the training data and places those in a resampled dataset. Some training examples won't show up in the bootstrapped collection, and some of these samples will be duplicates [14]. This results in a resampled dataset that has a distribution similar to the training set [9],

allowing for the computation of performance measures using the bootstrapped collection, which might be used as unique data.

Nevertheless, this approach only makes use of examples on which the model has previously been trained. Thus, not all classifiers will be able to use it. For instance, clustering algorithms will be trained to cluster together similar samples. Replicas, by default, will have a zero distance and will be clustered together. Hence, using a resampled set won't result in the addition of any additional info [14].

0.632+ method Borra et al. [2] make use of the 0.632+ bootstrap estimation developed by Efron and Tibsharini to evaluate the performance of the non-parametric bootstrap. The estimator is based on the theory that there will be about 0.632n instances of the original data in the resampled dataset.

The estimation accounts for the degree of overfitting that occurs in the error rate. We must therefore first calculate the relative overfitting ($\hat{R}$) before we can compute the estimation. The value for this ranges from 0 to 1, with 0 indicating no overfitting and 1 indicating significant overfitting. This is calculated by using the apparent error, the loss on all couples which is given by

$$\hat{y} = \frac{1}{n^2}\sum_{i,j} L(y_i, \hat{f}(x_j)), \tag{1}$$

and the leave-one-out bootstrap error rate. The leave-one-out bootstrap error is calculated by using the cases that were not included in the resampled set as a test sample:

$$err^{Bt} = \frac{1}{M}\sum_{j=1}^{M} + err_j^{Bt} \tag{2}$$

These equations are used to define the relative overfitting, which is the difference between the leave-one-out error and the apparent error divided by the difference between the loss and the apparent error,

$$\hat{R} = (err^{Bt} - err)/(\hat{y} - err) \tag{3}$$

The ultimate definition of the estimator will be a combination of the apparent error and the leave-one-out bootstrap error. The weight assigned to these errors will determine the ratio at which these will appear in the final estimation. If there is no overfitting, the weight is 0.632; if there is significant overfitting, the weight is 1. Using relative overfitting, the weight is determined:

$$\hat{w} = 0.632/(1 - 0.368 \cdot \hat{R}) \tag{4}$$

We can calculate the final estimation err^{B632+} given the weight. This will include a portion of both the apparent error and the leave-one-out bootstrap error. Hence, the bias caused by the apparent error will be compensated for by this error. This gives

$$err^{B632+} = \hat{w} \cdot err^{Bt} + (1 - \hat{w}) \cdot err \tag{5}$$

3.2.2 Parametric Bootstrap

The parametric model assumes that the data come from a distribution with unknown parameters, but that these parameters might be found. It looks for model parameters that fit the available data [6]. An estimated distribution will result from this. We can sample data points from this estimated distribution to create a resampled set. When tuning a model or determining how well a model is performing, this resampled set can be seen as new examples [14].

This bootstrap approach do have certain drawbacks, though. For example, determining the variance is difficult with a small collection of samples (less than 20 samples). Further, it can be challenging to decide how to deal with outliers, whether they might affect the final samples, and whether the output has to be smoothed [9]. This could lead to bias in the sampled data.

Parametric Bootstrap Procedure To assess the performance of parametric bootstrapping did Borra et al. [2] use a variation of the Efron-introduced Parametric Bootstrap Procedure. The objective is to take into account that the estimations are often to optimistic. The final estimator is the apparent estimation that has been corrected.

1. They apply a non-parametric bootstrap to obtain preliminary estimates of the parameters μ and σ^2. These parameters will desrcibe the estimated distribution.
2. They compute the density of the distribution. This can de computed with the estimated parameters μ and σ^2,

   $$\hat{f} = N(\hat{\mu}, \hat{\sigma}^2 I) \tag{6}$$

   For each datapoint in the dataset, they sample new values from the estimated distribution. For each original datapoint, B new values will be generated in total. The values are given by y_i^{*b}. With these revised values, a new μ will be calculated, $\hat{\mu}_i^{*b}$. A nonlinear smoothing function, $g()$, is used to calculate this with the generated values as input: $\hat{\mu}_i^{*b} = g(y_i^{*b})$.

   The covariance between each data point serves as the final estimator's corrector. Therefore it is first needed to estimate the covariance between y_i and $\hat{\mu}_i$. This is done by first computing the averaged datapoint,

   $$\bar{y}_i^{*} = \sum_b \frac{y_i^{*b}}{B}, \tag{7}$$

   then we can estimate the covariance

   $$\widehat{cov}_i = \sum_{b=1}^{B} \hat{\mu}_i^{*b}(y_i^{*b} - \bar{y}_i^{*})/(B-1) \tag{8}$$

3. Finally, the apparent error corrected with the covariance between y_i and $\hat{\mu}_i$ provides the estimator:

   $$err^{PB} = err + \frac{2}{n}\sum_i \hat{cov}_i \tag{9}$$

3.2.3 Bias-Corrected Bootstrap

Bootstrap methods are typically combined with a bias correction to eliminate the bias in bootstrapped sample sets. In their study Donna Chen and Metthaw S. Fritz explored six of these possible corrections for bias on the mean.

Mean correction (z_{mean}) Given the estimated parameters that describe the distribution, the Efron and Tibshirani correction (z_0) is a z-score, which shows the percentage where the estimated distribution is less than the original distribution. The z_{mean} correction also takes the mean into the measure. The percentage of resampled points that are below the mean of the estimated distribution, instead of the original distribution, will be applied as a bias percentile correction. The bias percentile correction takes a percentage and establishes an upper and lower bound on an ordered set of data, allowing only this percentage of the data to remain. For example, only data larger than the datapoint at 5% and data smaller than the datapoint at 95% will be taken into account when doing the adjustment for 90% [3].

Winsorized mean (z_{win}) The Winsorized mean will take a percentage of the values on both sides of the distribution and trim them. Chen et al. provide a clear example. The 20% Winsorized Mean is determined using the values 1–10 by:

$$z_{win20} = \frac{2+2+3+4+5+6+7+8+9+9}{10} \tag{10}$$

The most extreme value is thus substituted by 2 and 9 respectively on both sides. It will be comparable to z_{mean} when the trim is set to zero. It is equivalent to the percentile bootstrap when 50% trim is present. Therefore only Winsorized means of 10%, 20%, 30%, and 40% are considered in the experiments [3].

Medcouple (z_{mc}) The medcouple focuses on the skewness of the data. It describes the distribution's skew., the medcouple is a standardized weighted median. The median is typically defined as,

$$m_n = \begin{cases} \frac{x_{n/2} + x_{n/2+1}}{2} \text{EVEN N} \\ x_{(n+1)/2} \text{ODD N}, \end{cases} \tag{11}$$

for all x_i yields: $x_i \leq x_j \;\&\; i < j$

To compute the medcouple, we first define $h(x_i, x_j)$. This will determine the variation in each pair's distance from the median,

$$h(x_i, x_j) = \frac{(x_j - m_n) - (m_n - x_i)}{x_j - x_i} \tag{12}$$

The division standardizes the outcome. Hence, $-1 \leq MC_n \leq 1$.

Taking into consideration this variation in distance the medcouple, the standarized median, can be computed:

$$MC_n = \underset{x_i \le m_n \le x_j}{med} h(x_i, x_j), \tag{13}$$

3.3 Cross-validation and Bootstrap combined Sampling Methods

Recently, attempts have been made to combine Cross-Validation and Bootstrap sampling into a single algorithm in order to make model validation more computationally efficient without compromising accuracy. Below we present two such methods.

3.3.1 Bootstrap Bias Corrected Cross-Validation

Bootstrap Bias Corrected Cross-Validation (BBC-CV) is a method proposed in [17] for efficient and accurate performance estimation. In order to explain how it works, let us first define the *configuration selection strategy* (css) function [17]. With Π defined as the matrix with the out-of-sample predictions of N rows and C columns, where N is the sample size and C is the number of configurations among which we wish to select the best one, $[\Pi]_{ij}$ denotes the out-of-sample prediction of on the i-th sample of the j-th configuration. The ground truth labels are denoted as y. The function *css* returns the index of the best-performing configuration according to some criterion. Using the minimum average loss as the criterion, css can be defined as:

$$css(\Pi, y) = argminl(y, \Pi(:, i))$$

BBC-CV utilizes the out-of-sample predictions Π generated by CVT, and then generates B bootstrapped matrices Π^b by randomly sampling N rows of Π with replacement. Additionally, corresponding vectors of true labels y_b are created for each bootstrapped matrix. The matrices $\Pi^{\setminus b}$, b = 1, . . . ,B are then formed by including the samples that were not used in Π^b (i.e., $\Pi \setminus \Pi^b$), and $y^{\setminus b}$ represents their corresponding vectors of true labels.

For each bootstrap iteration b, the BBC-CV method applies the configuration selection strategy css(Π^b, y^b) to select the optimal configuration i that yields the best performance, and subsequently computes the loss L_b of configuration i as L_b = l($y^{\setminus b}$,$\Pi^{\setminus b}$). Ultimately, the estimated loss *LBBC* is calculated as the average of L_b over all B bootstrap iterations.

3.3.2 Bootstrap Corrected with Early Dropping Cross-Validation

Bootstrap Corrected with Early Dropping Cross-Validation (BCED-CV) is an approach that uses heuristic procedures to improve the computational efficiency of BBC-CV through means of early stopping. The out-of-samples predictions from CV are used in a statistical hypothesis test that determines if a configuration's performance is significantly worse than the performance of the current best configuration. In such a scenario, the underperforming configuration can be discarded early on, and no further models will be trained under that configuration in subsequent folds. The statistical test operates under the null hypothesis that a specific configuration's performance is equivalent to the current best configuration. This hypothesis is assessed for each configuration that remains in consideration at the conclusion of each fold. The current out-of-sample predictions for all configurations still being evaluated are utilized to identify the best configuration during the test. By employing bootstrapped matrices, we can determine the likelihood of a particular configuration exhibiting worse performance. This probability is estimated by calculating the percentage of times the configuration's loss is higher than that of the best configuration. If this probability exceeds a certain significance threshold, the configuration is eliminated from further consideration.

4 Comparison and Discussion

In this section we describe the experiments that were executed by various researchers and present their findings.

4.1 BBC-CV and BCED-CV evaluation against Cross-Validation variants

Tsamardinos, Greasidou, Tsagris and Borboudakis [17] performed a set of experiments in order to compare the performance and computational efficiency of the methods CVT, TT, NCV, BBC-CV and BCED-CV, both on synthetic and real-world data.

4.1.1 Synthetic Datasets

Experimental setup The authors constructed several synthetic datasets using four different Beta distributions with varying sample sizes $N \in \{20, 40, 60, 80, 100, 500, 1000\}$. They ran the experiments with number of configurations to compare $C \in \{50, 100, 200, 300, 500, 1000, 2000\}$, resulting in 196 different experimental settings. The number of bootstraps was set to 1000 for both BBC-CV and BCED-CV, while the dropping threshold for BCED-CV was set to 0.99. The data was split into 10 folds in the same way for all protocols. The internal loop of NCV uses 9 folds. The performance is judged based on the accuracy metric and the results they reported were the average over 500 runs for each setting.

Results To evaluate each method, the bias is used, which is defined as the difference between the estimated and true performance of a configuration. If there is a positive bias, it means that the actual performance is lower than what was estimated by the performance estimation protocol. This implies that the protocol is optimistic and overestimates the performance. On the other hand, if there is a negative bias, it means that the estimated performance is conservative.

In all settings, the CVT estimate of performance is biased in an optimistic direction. As the sample size decreases, the bias of CVT tends to overestimate the performance of the final model. Conversely, as the sample size increases, the bias of CVT tends towards zero. The bias of the CVT estimate also increases as the number of models under comparison increases, although this effect is relatively small in this experiment. For small sample sizes (≤ 100), the behavior of TT varies greatly and is highly sensitive to the number of configurations, while for larger sample sizes (≥ 500), TT is systematically conservative, over-correcting the bias of CVT. NCV provides an almost unbiased estimation of performance across all sample sizes, but is computationally expensive since the number of models that need to be trained depends quadratically on the number of folds K. BBC-CV provides conservative estimates, with low bias that quickly tends towards zero as the sample size increases. Compared to TT, it is better fitting for small sample sizes and produces more accurate estimates overall. In comparison to NCV, BBC-CV is somewhat more conservative. However, the much lower computational cost of BBC-CV (one order of magnitude) compensates for its conservatism. BCED-CV displays similar behavior to BBC-CV, with lower bias that approaches zero faster. It is on par with NCV, being slightly more biased. Furthermore, BCED-CV is up to one order of magnitude faster than CVT and two orders of magnitude faster than NCV.

4.1.2 Real datasets

Experimental setup These methods were also evaluated on real datasets, obtained from popular data science challenges. The datasets used are: christine, jasmine, phlippine, madeline, sylvine, gisette, madelon, dexter and gina. To create the experimental datasets, each original dataset D was divided into two stratified subsets, D_{pool} and $D_{holdout}$. D_{pool} was composed of 30% of the total samples in D, while $D_{holdout}$ was composed of the remaining 70% of the samples. With the exception of the dexter dataset, 20 sub-datasets were created for each sample size N $\in \{20, 40, 60, 80, 100, 500\}$ by sampling (without replacement) from D_{pool}. For the dexter dataset, 20 sub-datasets were created for each N $\in \{20, 40, 60, 80, 100\}$. Overall, 1060 sub-datasets were generated. The true performance of the final, selected model of each of the tested protocols was estimated using $D_{holdout}$. The metric of performance that was chosen was the AUC with respect to the bias of the model. Random forests, SVMs [4] and LASSO [15] were the learning algorithms of choice, each applied with various sets of hyper-parameters. These hyper-parameter in combination with the different

methods of preprocessing and feature selection produced a total of 610 configurations. The parameters for the BBC-CV and BCED-CV methods are kept the same as in the synthetic dataset experiments.

Results Once again, the bias is used for evaluation purposes. Even though there is some variation among the different datasets, the results of these experiments are in agreement with those of the simulation studies as presented in 4.1.1. These findings show that BBC-CV and BCED-CV outperform other methods, such as Nested Cross-Validation and the TT method. This is achieved by either providing more precise, almost unbiased, or conservative estimates of performance, even for smaller sample sizes, and/or having significantly lower computational costs. Thus, instead of NCV, the authors suggest the use of BBC-CV for small sample sizes or BCED-CV for larger sample sizes.

4.2 Bootstrapping protocols

We will examine two publications that assessed the feasibility of the bootstrapping methods in order to compare their performance. The comparison of non-parametric and parametric bootstrap will be done first, followed by a comparison of various bias corrections.

4.2.1 Parametric vs Non-Parametric

We will use the publication 'Measuring the Prediction Error', written by Borra et al., to compare the parametric and non-parametric bootstrap. With regard to various types of errors, it analyzes parametric and non-parametric bootstrap approaches.

Experimental setup Borra et al. conducted two separate tests to assess how well both bootstrap estimation predictors performed [2]. The simulations are performed using regression functions that have four or five covariates, respectively. Because those variables were chosen at random, the distribution was also random. So, this is artificial data. Three alternative techniques were used to estimate the regression function. Using neural networks(NN), projection pursuit regression(PPR), and regression trees(RT). The estimated function is used to compute the forecasted outcome. In order to determine the so-called extra-sample error, this value will be contrasted with the true value.

We can calculate the relative bias (rb_h) and the mean absolute relative bias($\bar{arb}$) using this info. An unbiased estimator has rb=0 for each h, resulting in a low value for arb. The bootstrap methods considered are the Bootstrap 0.632+ estimator with 100 bootstrap examples and the parametric bootstrap estimator with 100 bootstrap samples.

Results The experiment's findings are displayed in tables 1 and 2. The parametric approach outperforms the non-parametric method in both experiments. It is obvious that the parametric estimator is generally far more accurate. Table 2, however, demonstrates that when there is little noise, the non-parametric approach is more accurate. Data noise is expected in the actual world, hence, for the estimator to be usable for real data, it must perform well at higher noise ratios.

	TR		PPR		NN	
	Sample size		Sample size		Sample size	
$\bar{arb}$	120	500	120	500	120	500
Parametric	0.074	0.045	0.062	0.012	0.057	0.010
Non-Parametric	0.056	0.055	0.182	0.082	0.080	0.022
$\bar{rse}$						
Parametric	1.956	2.485	1.183	0.945	0.948	0.780
Non-Parametric	1.732	2.669	2.185	1.961	1.206	1.079
comparative performance*						
Parametric	3.67	3.40	3.30	2.75	2.71	2.45
Non-Parametric	3.13	3.86	5.14	5.39	3.68	3.60

Table 1. The table was created using the data from a table that was presented in 'Measuring the Prediction Error' [2] (simulation A) — **An estimator mean rank, lower values indicate better estimators*

4.2.2 Evaluation of Bias-Corrected Bootstrap

Experimental setup To compare various bias corrections, Chen et al. performed numerous experiments [3]. The original bias-correction (z_0) was contrasted with the z_{mean}, various z_{win}, and z_{mc}. They produced observations and residuals using the R function

	TR			PPR			NN		
	Ratio noise			Ratio noise			Ratio noise		
$\bar{arb}$	1	2.5	5	1	2.5	5	1	2.5	5
Parametric	0.044	0.039	0.039	0.015	0.024	0.053	0.016	0.018	0.017
Non-Parametric	0.023	0.049	0.051	0.047	0.099	0.160	0.022	0.029	0.049
$\bar{rse}$									
Parametric	2.20	1.99	1.82	1.76	0.83	0.67	1.54	0.72	0.53
Non-Parametric	1.74	2.11	2.00	2.34	1.63	1.57	2.18	1.04	0.89
comparative performance*									
Parametric	4.31	3.68	3.54	3.85	3.04	2.97	3.49	2.84	2.56
Non-Parametric	3.38	4.08	4.03	5.01	5.84	6.31	3.84	3.70	3.97

Table 2. The table was created using the data from a table that was presented in 'Measuring the Prediction Error' [2] (simulation B)

rnorm(). So, they used synthetic data. All approaches were used after sampling the data 1000 times. This entire process was repeated a total of 1000 times.

Also, the effectiveness is evaluated using information from the real dataset Athletes Training and Learning to Avoid Steroids program (ATLAS). We won't talk about those findings though because they don't imply much in a broader sense.

Used terms There will be an introduction to the many terms that will be utilized in the comparison. Errors of two types are present. Errors of Type I and Type II. Type I errors are regarded as false negatives because they reject a null hypothesis that is correct. Type II errors are characterized as false negatives because they keep a hypothesis while it is actually untrue [10]. Both of the errors ought to be minimal. In this context, a significant Type I mistake suggests that there is a large likelihood that the correction would lead to false negative results.

The word imbalance is used in the comparison. Unevenly distributed datasets are known as imbalanced datasets. The term statistical power is also frequently used. The statistical power is between 0 and 1. It represents the possibility to have a true positive. It only has true positives when the value is 1. Its maximum Type II mistakes occur when it is 0. It is a way to detect if a difference between test variations genuinely exists [5].

The term effect size is used. This term quantifies how closely related a dataset's instances are to one another. The more distinct the differences between the samples are based on their features, the larger the effect size. The authors of the utilized paper use a robustness range in their comparison. This is the range in which results are error-resistant [3]. They are resistant to outliers, for instance.

Results In their experiments, Chen et al. found that z_0, z_{mc}, and z_{mean} have extremely similar Type I errors. They found that the errors of the corrections frequently fall within the robustness range or fall outside of it altogether. They observe that the Type I error grows with increasing trimming for the Winsorized means. Even so, compared to z_{win40}, the percentile bootstrap often displays lower error rates.

The statistical power rises with the size of the sample for all corrections. The powers of z_0, z_{mc}, and z_{mean} were similar. Depending on the parameter combination, one may be more powerful than another. Chen et al. found that as the level of trimming grew, the power of the Winsorized means declined [3]. Overall, z_{mean} has the greatest power, whereas z_{win40} exhibits the least. All approaches have a propensity to be imbalanced in the same way.

To sum up, the Type I error rates and power of the medcouple and mean are comparable to those of the BC bootstrap. If the trimming is increased, the Type I error and power for the Winsorized means will drop. As opposed to the percentile bootstrap, it exhibits greater power because outliers are not eliminated but rather given another value.

The findings also demonstrate the close relationship between Type I error and power. It is important to take the other one into account when evaluating one of them. Also, the experiments showed that it is increasingly likely that the Type I error and power will reach optimal levels as the sample and effect size grow.

The benchmark approaches (Monte Carlo and Percentile Bootstrap, not specifically covered in this paper), according to Chen et al. [3], have the most reliable robust Type I error rates. The error rates for the other approaches will be robust when the sample size is more than 250, which also produced the best power and coverage. Consequently, when the sample size is less than 250, the effects of the various approaches are most noticeable.

Consequently, they advise using a benchmark approach when there should be minimal Type I error and when the sample size is small. The BC Bootstrap, which has the most power, should be used if researchers are concerned about the Type II mistake. Use the Winsorized means, which specifies the field between, to find a trade-off. A Winsorized mean with a low percentage (10%) is more comparable to BC Bootstrap, whereas one with a high percentage (40%) is more comparable to the Monte Carlo approach. Hence, a compromise between Type I and power will result from a percentage in the middle [3].

4.3 General comparison of CV and Bootstrap

Both Bootstrapping and Cross-Validation and variants of them are being widely used and have been thoroughly studied. Comparisons between them have been made by independent research groups, without, however, agreeing and indicating one of them as the objectively better technique. Kohavi [11] in his experiments with decision trees and Naive-Bayesian classifiers found that using stratified ten-fold cross-validation worked best, while for Borra S. and Di Ciaccio A. [2] a variation of Cross-Validation and the Parametric Bootstrap yielded the best results. Ljumovic M. and Klar M. [13] found that both techniques provided equally good results in their experiments with random forest classifiers. Tsamardinos, Greasidou, Tsagris and Borboudakis [17], on the other hand, proved with their simulations and experiments on real data that a novel approach that combines both methods (BBC-CV) is also competitive.

These findings enforce the notion that the best method for performance estimation and model selection is not universal but varies and depends on the dataset and learning algorithms used. Empirically, CV is the standard procedure to be followed in the general case, but bootstrapping should be preferred when confidence intervals are important or when basic assumptions regarding the dataset are violated. One of these assumptions the sample size; if it is too small, the estimations provided by bootstrap sampling based methods will be more accurate than those of CV. Another ”violation” is inaccurate or unreliable data, i.e when the practitioner cannot be sure that the dataset originates from a trusted source and that the noise contained in it is low enough. Lastly, ideally the dataset consists of identically distributed data points, which means that all the datapoints are taken from the same probability distribution and the distribution does not fluctuate.

5 Conclusion

Extensive research has been carried out on sampling methods for the validation of machine learning models in order to determine the best one. In this paper we collected a series of comparative studies on Cross-Validation and Bootstrap based techniques. The study aimed to provide more guidance than is currently provided in the literature when faced with the huge variety of sampling method options. We provided an overview of these techniques, discussed their advantages and disadvantages and we propagated the results of these studies, which are not aligned on the preferred method. The method that yields the best results changes on a per-case basis, indicating that the main question we are tackling does not have a simple definitive answer. As a rule of thumb, Cross-Validation and variants of it are recommended in a general sense, unless the user is dealing with irregular data, in which case Bootstrap methods may provide better estimations.

Approach assessment The method used to choose the papers does not guarantee that the most recent and widely used sampling techniques have been taken into account. Nevertheless, our research tries to present an overview of several methodologies, the paper does provide insights regarding those approaches. The method of comparison is now solely founded on prior works. We did not conduct the experiments that would have allowed us to compare different approaches that are based on different fundamental principles. As such, it was difficult to provide a fair cross-paper comparison between the various methods examined by different research teams, considering the dissimilarities between the experiments the algorithms were evaluated on and the metrics used to gauge their performance.

Future work A possible line of future work on this topic includes a large scale experiment in which all the aforementioned methods are compared against each other under the same experimental conditions and on a number of metrics. The results of this experiment could help paint a more accurate picture about the strengths and weaknesses of each method. Additionally, tests can be done to determine which factors affect the choice of sampling method the most. In this paper we show that the nature of the dataset and the family of models used lead to different best methods, but further research can delve deeper and show for each model family, such as Linear Regression or Neural Networks, which method is most suitable.

Acknowledgements

The authors would like to thank Michael Biehl for bringing up this subject and offering insightful feedback. Also, we appreciate the careful review of our paper by other students. Finally, we would like to express our gratitude to the Student Colloquium team for their lectures and support during the course.

References

[1] D. D. Boos. Introduction to the bootstrap world. *Statistical Science*, 18(2), 2003.

[2] S. Borra and A. Di Ciaccio. Measuring the prediction error. a comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics &; Data Analysis*, 54(12):2976–2989, 2010.

[3] D. Chen and M. S. Fritz. Comparing alternative corrections for bias in the bias-corrected bootstrap test of mediation. *Evaluation &; the Health Professions*, 44(4):416–427, 2021.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] B. P. Cross and P. C. P. C. is an experienced marketing consultant who specializes in conversion optimization. Statistical power: What it is and how to calculate it in a/b testing, Dec 2022.

[6] A. C. Davison and D. V. Hinkley. Bootstrap methods and their application. 1999.

[7] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1979.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[9] D. Hinkley. [bootstrap: More than a stab in the dark?]: Comment. *Statistical Science*, 9(3), 1994.

[10] W. Kenton. Type 1 error: Definition, false positives, and examples, Feb 2023.

[11] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, page 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[12] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14, 03 2001.

[13] M. Ljumović and M. Klar. Estimating expected error rates of random forest classifiers: A comparison of cross-validation and bootstrap. In *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, pages 212–215, 2015.

[14] D. of Statistics Online Programs The Pennsylvania State University. 15.3 - bootstrapping. Online; accessed 22 February 2023.

[15] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[16] R. J. Tibshirani and R. Tibshirani. A bias correction for the minimum error rate in cross-validation. *The Annals of Applied Statistics*, 3(2):822 – 829, 2009.

[17] I. Tsamardinos, E. Greasidou, M. Tsagris, and G. Borboudakis. Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation, 2017.

[18] J. Wainer and G. C. Cawley. Nested cross-validation when selecting classifiers is overzealous for most practical applications. *CoRR*, abs/1809.09446, 2018.

university of groningen / faculty of science and engineering / computing science