WEST UNIVERSITY OF TIMIŞOARA
DOMAIN: COMPUTER SCIENCE

HABILITATION THESIS

CANDIDATE:

Mădălina Eraşcu
Associate Professor, PhD
West University of Timişoara
Faculty of Mathematics and Computer Science
Department of Computer Science

TIMIŞOARA
2023

WEST UNIVERSITY OF TIMIŞOARA
DOMAIN: COMPUTER SCIENCE

Formal Methods Supported by Symbolic Computation for Engineering Applications in Cloud Computing and Artificial Intelligence

CANDIDATE:

Mădălina Eraşcu
Associate Professor, PhD
West University of Timişoara
Faculty of Mathematics and Computer Science
Department of Computer Science

TIMIŞOARA
2023

Abstract

This habilitation thesis presents the most significant scientific and academic achievements of its author, beginning in December 2012 when she defended her PhD thesis titled *Computational Logic and Quantifier Elimination Techniques for (Semi-)automatic Static Analysis and Synthesis of Algorithms* at the Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria. During this period, the author *collaborated* with established scientists as well as junior researchers. The resulted work, submitted to conferences and journals, is highly relevant in the fields of *formal methods, automated deduction, and symbolic computation*, with applications in *Cloud Computing* and *Artificial Intelligence.*

The author also served as the director of two national competitive research grants: *MANeUveR: Management Agency for Cloud Resources* and *SAGE: A Symbiosis of Satisfiability Checking, Graph Neural Networks, and Symbolic Computation.*

Additionally, she was awarded the prestigious *Fulbright-RAF Scholar Award in Entrepreneurship,* which emphasizes her interests in entrepreneurship education for Computer Science students, and not only, and in exploring the market potential of research results.

The author started her *scientific career* during her MSc studies at the Research Institute for Symbolic Computation. She began *teaching activities* in February 2014 at the Department of Computer Science, West University of Timisoara, initially as an Associate Assistant and later as a Lecturer. On October 2021, she was promoted to Associate Professor. At West University of Timisoara, *she introduced courses on formal methods and satisfiability checking* at both the Master's and Bachelor's levels. Notably, *she integrated research topics into her teaching activities*, exposing Computer Science students to research early in their studies. Most of the Bachelor's and Master's theses she supervised were research-oriented. All these achievements are detailed in Chapter 2 of the thesis.

Chapter 3 of the thesis provides a comprehensive overview of the candidate's research contributions, with a *focus on formal methods in various domains, including symbolic computation, Cloud computing and data-intensive applications.* It also highlights *contributions to machine learning and computer science education*, which serve as preliminary work for applying formal methods in these exciting domains and, respectively, developing effective methods for teaching formal methods.

Section 3.1 centers on *formal methods combined with symbolic computation*, showcasing the candidate's significant work on real quantifier elimination for synthesizing optimal numerical algorithms, particularly in the case study of square root computation. The section further highlights efficient simplification techniques for special real quantifier elimination, with applications to the synthesis of optimal numerical algorithms.

Section 3.2 discusses the *utilization of formal methods in Cloud Computing*, with a particular focus on Security SLA enforcement and component-based application deployment optimization. Topics covered include automated security SLA enforcement in the Cloud, a methodology for setting up security capabilities on Cloud services, and benchmarking optimization solvers for efficient deployment. The section provides insights into experimental analysis, showcasing results and discussions

on solver performance.

Section 3.3 presents the work on *formal verification and quality assessment of distributed systems in the context of Storm technology.* It introduces an automated formal verification approach utilizing Storm topologies' formal models constructed through CLTLoc metric temporal logic, reinforced by counters. Formal models are generated from high-level topological descriptions, while the Zot verification tool assesses queue-related properties. Complementary to the CLTLoc metric temporal logic approach, it formalizes data-intensive applications on Storm through an array-based systems formalism, utilizing quantified first-order logic and the Cubicle model checker to verify safety properties.

Section 3.4 delves into various applications and methodologies. It begins by presenting a *tool for fake news detection*, addressing the problem statement and the approach taken. The implementation details, including parsing, machine learning, and cosine similarity techniques, are outlined. The experimental results showcase the tool's effectiveness in content and title detection. Additionally, this section covers the *architectural development of Binarized Neural Networks (BNNs) for traffic sign recognition.* It explains BNNs, datasets, and the experimental setup. The proposed methodology, including XNOR and Binarized Neural Architectures, is discussed in detail, as well as the experimental results.

Section 3.5 centers on *Computer Science Education*, delving into the research conducted on evaluating a student-centered learning approach through the lens of computer science students.

The thesis concludes with Chapter 4, which outlines the scientific and academic roadmap of the author.

Rezumat

Această teză de abilitare prezintă cele mai semnificative realizări științifice și academice ale autoarei sale, începând din decembrie 2012, când aceasta și-a susținut teza de doctorat intitulată *Logica Computațională și Tehnici de Eliminare a Cuantificatorilor pentru Analiza și Sinteză Statică (Semi-)automată a Algoritmilor* la Institutul de Cercetare pentru Calcul Simbolic, Universitatea Johannes Kepler, Linz, Austria. În timpul acestei perioade, autoarea a colaborat cu cercetători consacrați, precum și mai tineri pentru a produce rezultate care au fost prezentate la conferințe și publicate în reviste. Aceste contribuții sunt extrem de relevante în *domeniile metodelor formale, deducției automate și calculului simbolic*, cu aplicații în *Cloud Computing* și *Inteligență Artificială.*

Autoarea a fost, de asemenea, director al două granturi de cercetare competitive naționale: *MANeUveR: Agenția de Management pentru Resursele Cloud* și *SAGE: O Simbioză a Verificării Satisfiabilității, Rețelelor Neurale Grafice și Calculului Simbolic.*

În plus, i s-a acordat prestigiosul premiu *Fulbright-RAF Scholar Award în Antreprenoriat*, ceea ce subliniază interesul său pentru *educația antreprenorială a studenților* și *explorarea potențialului de piață al rezultatelor cercetării.*

Autoarea și-a început *cariera științifică* în timpul studiilor de master la Institutul de Cercetare pentru Calcul Simbolic. *Activitățile de predare* le-a început în februarie 2014 la Departamentul de Informatică al Universității de Vest din Timișoara, inițial ca asistent asociat, iar mai târziu ca lector. În octombrie 2021, a fost promovată la gradul de conferențiar. La Universitatea de Vest din Timișoara, *a introdus cursuri despre metodele formale și verificarea satisfiabilității* la nivel de master și licență. De remarcat este *integrarea subiectelor de cercetare în activitățile sale de predare*, expunând studenții de informatică la cercetare încă din primele etape ale studiilor lor. Cea mai mare parte dintre tezele de licență și master pe care le-a coordonat au avut un caracter de cercetare. Toate aceste realizări sunt detaliate în Capitolul 2 al tezei.

Capitolul 3 al tezei oferă o prezentare cuprinzătoare a contribuțiilor de cercetare ale candidatului, cu accent pe *metodele formale* în diverse domenii, inclusiv *calculul simbolic, Cloud computing și aplicații de prelucrare intensivă a datelor.* De asemenea, se evidențiază contribuțiile la *învățarea automată* și *educația în informatică*, care servesc drept lucrări preliminare pentru aplicarea metodelor formale în aceste domenii interesante sau pentru dezvoltarea de metode eficiente de predare a metodelor formale.

Secțiunea 3.1 se concentrează asupra *metodelor formale combinate cu calculul simbolic*, prezentând rezultatele candidatului privind eliminarea cuantificatorilor logici în vederea sintezei algoritmilor numerici optimi, în special în studiul de caz privind calculul rădăcinii pătrate. Secțiunea evidențiază în continuare tehnici eficiente de simplificare pentru eliminarea cuantificatorilor, cu aplicații în sinteza algoritmilor numerici optimi.

Secțiunea 3.2 discută utilizarea *metodelor formale în Cloud Computing*, cu accent pe aplicarea Service Level Agreements (SLAs) în Securitate și optimizarea implementării aplicațiilor bazate pe componente. Sunt abordate subiecte precum aplicarea automată a SLAs pentru securitate în Cloud, metodologie pentru configurarea capacităților de securitate pe serviciile Cloud și evaluarea performanțelor

soluțiilor de optimizare. Secțiunea oferă informații privind analiza experimentală, prezentând rezultate și discuții privind performanța soluțiilor de optimizare.

Secțiunea 3.3 prezintă lucrările privind *verificarea formală și evaluarea calității sistemelor distribuite în contextul tehnologiei Storm.* Aceasta introduce o abordare automată de verificare formală utilizând modele formale ale topologiilor Storm construite prin logica temporală metrică CLTLoc. Modelele formale sunt generate din descrieri topologice la nivel înalt, în timp ce instrumentul de verificare Zot evaluează proprietățile legate de cozi. Ca și complement la această abordare, se formalizează aplicațiile de prelucrare intensivă a datelor pe platforma Storm printr-un formalism bazat vectori, utilizând logica de ordinul întâi și verificatorul de model Cubicle pentru a verifica proprietăți de siguranță.

Secțiunea 3.4 explorează diverse aplicații și metodologii. Ea începe prin prezentarea unei *unelte pentru detectarea știrilor false*, abordând enunțul problemei și abordarea adoptată. Sunt detaliate aspectele implementării, inclusiv tehnici de parsare, învățare automată și tehnici de similaritate cosinus. Rezultatele experimentale demonstrează eficacitatea uneltei în detectarea conținutului și a titlurilor false. În plus, această secțiune acoperă *dezvoltarea arhitecturală a Rețelelor Neurale Binarizate* pentru recunoașterea semnelor de circulație. Metodologia propusă, inclusiv arhitecturile XNOR și Rețelele Neurale Binarizate, este discutată în detaliu. La fel și rezultatele experimentale și implicațiile acestora.

Secțiunea 3.5 se concentrează pe *Educația în Informatică*, explorând cercetarea efectuată pentru evaluarea unei abordări de învățare centrată pe student prin ochii studenților de informatică.

Teza se încheie cu Capitolul 4, care conturează traseul științific și academic al autorului.

Contents

Abstract . . . i
Rezumat . . . iii

1 Introduction 1
1.1 Overview . . . 1
1.2 List of Main Contributions . . . 3

2 Scientific and Professional Career: Past and Present 5
2.1 Important scientific achievements of the candidate . . . 5
2.2 Fundraising and coordination of research teams . . . 7
2.3 Facilitating research among students . . . 8
2.4 Other research related achievements . . . 8
2.5 Teaching activities . . . 9

3 Presentation of the Research Contributions against State-of-the-art 11
3.1 Formal Methods in Symbolic Computation . . . 11
3.1.1 Real quantifier elimination for the synthesis of optimal numerical algorithms (Case study: Square root computation) . 12
3.1.2 Efficient Simplification Techniques for Special Real Quantifier Elimination with Applications to the Synthesis of Optimal Numerical Algorithms . . . 19
3.1.3 Conclusions . . . 22
3.2 Formal Methods in Cloud Computing . . . 22
3.2.1 Automatically Enforcing Security SLAs in the Cloud . . . 23
3.2.2 A Security SLA-driven Methodology to Set-up Security Capabilities on Top of Cloud Services . . . 24
3.2.3 Benchmarking Optimization Solvers and Symmetry Breakers for the Automated Deployment of Component-based Applications in the Cloud . . . 25
3.2.4 Conclusions . . . 31
3.3 Formal Methods for Data-intensive Applications . . . 31
3.3.1 Towards the Formal Verification of Data-Intensive Applications Through Metric Temporal Logic . . . 33
3.3.2 Formal verification of data-intensive applications through model checking modulo theories . . . 35
3.3.3 D-VerT: a tool for verification of big-data applications . . . 39
3.3.4 Conclusions . . . 41
3.4 Machine Learning . . . 42

3.4.1 A Tool for Fake News Detection 42
3.4.2 Architecturing Binarized Neural Networks for Traffic Sign Recognition 49
3.4.3 Conclusions 55
3.5 Computer Science Education 56
3.5.1 Transferring Learning into the Workplace: Evaluating a Student-Centered Learning Approach through Computer Science Students' Lens 57
3.5.2 Conclusions 65

4 Scientific and Professional Roadmap 67
4.1 Envisaged Scientific and Professional Endeavors 67
4.2 Upcoming Research in the Near Future 68
4.3 Importance 68
4.4 Difficulties 69
4.5 Limitations of the state-of-the-art 69
4.6 Elements of originality and innovation in relation to the state-of-the-art 71

Bibliography 72

Chapter 1

Introduction

1.1 Overview

This habilitation thesis is the result of my research in the vast area of *artificial intelligence*, both continuing the topics investigated during the PhD studies and approaching new topics. After my PhD defense, I have continued working at the intersection of formal methods and symbolic computation. Simultaneously being part of the implementation team of two EU funded projects, I had the opportunity explore how formal methods can be applied to domains like Cloud Computing and Data-intensive Applications. Motivated by the magnitude of the fake news phenomenon, I have started a project on fake news detection. The short experience in automotive industry as well as the well-known shortcomings of machine/deep learning determined the research on binarized neural networks and their formal analysis. Having introduced topics on formal methods into my teaching, I was confronted with the dilemma how to efficiently teach and evaluate. This led to an experiential learning approach which was validated by established methods from educational sciences. In the remaining section of this chapter we present our main contributions in the areas mentioned above including the degree of involvement in each of the topics covered by this thesis (Section 1.2).

Chapter 2 outlines the scientific and professional journey of the candidate, spanning from the past to the future. It delves into the candidate's accomplishments, including significant scientific achievements, fundraising, research team coordination, and facilitation of student research. It also highlights other research-related achievements and teaching activities undertaken.

The rest of the thesis presents a comprehensive overview of the research contributions of the candidate, with a *focus on formal methods in various domains* (Sections 3.1–3.3): *symbolic computation*, *Cloud computing*, and *data-intensive applications.* Additionally, contributions to *machine learning* and *computer science education* are presented in Sections 3.4 and 3.5 which nonetheless are preliminary works for applying formal methods in this exciting domain, respectively, find methods helping effectively teach formal methods.

Section 3.1 centers on *formal methods combined with symbolic computation*, showcasing the candidate's significant work on real quantifier elimination for synthesizing optimal numerical algorithms, particularly in the case study of square root computation. The section further highlights efficient simplification techniques for special real quantifier elimination, with applications to the synthesis of optimal numerical algorithms.

Section 3.2 discusses the utilization of *formal methods* in *Cloud Computing*, particularly focusing on Security SLA enforcement and component-based application deployment optimization. It covers topics such as automated security SLA enforcement in the Cloud, a methodology for setting up security capabilities on Cloud services, and benchmarking optimization solvers for efficient deployment. The section provides insights into experimental analysis, showcasing results and discussions on solver performance.

Section 3.3 presents the work on formal verification and quality assessment of distributed systems in the context of Storm technology. It introduces an automated formal verification approach utilizing Storm topologies' formal models constructed through CLTLoc metric temporal logic, bolstered by counters. Formal models are generated from high-level topological descriptions, while the Zot verification tool assesses queue-related properties. Complementary to the CLTLoc metric temporal logic approach, it formalizes data-intensive applications on Storm through array-based systems formalism, utilizing quantified first-order logic and Cubicle model checker to verify safety properties. These approaches have been implemented in the DICE Verification Tool (D-VerT) within the DICE framework, hence, enabling comprehensive quality assessment and verification for Storm topologies using CLTLoc metric temporal logic and array-based systems formalisms.

Section 3.4 delves into various applications and methodologies. It commences by presenting a tool for fake news detection, addressing the problem statement and the approach taken. The implementation details, including parsing, machine learning, and cosine similarity techniques, are outlined. The experimental results showcase the tool's effectiveness in content and title detection. Additionally, this section covers the architectural development of Binarized Neural Networks (BNNs) for traffic sign recognition. It explains BNNs, datasets, and the experimental setup. The proposed methodology, including XNOR and Binarized Neural Architectures, is discussed in detail. The section concludes with a presentation of experimental results and discussions surrounding their implications.

Section 3.5 centers on *Computer Science Education*, delving into the research conducted on transferring learning into the workplace and evaluating a student-centered learning approach through the lens of computer science students. It discusses the design and aim of the study, research context, learning transfer into the workplace of the student-centered learning initiative, methodology, participant characteristics, measures, data collection, data analysis, results, and conclusions.

Chapter 4 presents a roadmap for the candidate's advancement and growth in their scientific and professional pursuits. This roadmap entails envisioned scientific and professional endeavors that are expected to be undertaken in the future. Additionally, it explores upcoming research in the near future, emphasizing its importance, potential difficulties, and the limitations of the existing state-of-the-art. The roadmap also examines the elements of originality and innovation in relation to the current state-of-the-art, indicating the candidate's commitment to pushing the boundaries of knowledge and contributing to their field.

1.2 List of Main Contributions

This section outlines the candidate's noteworthy achievements in their research and academic journey. These contributions encompass original *scientific* methodologies, theoretical advancements, empirical findings, and practical applications that have significantly advanced their field's understanding. Also, it highlights the candidate's impact on *academia*, demonstrating their ability transfer knowledge and expertise to the next generation of scholars and practitioners.

The contributions from Section 3.1 focus on applying real quantifier elimination to synthesize optimal numerical algorithms, with a case study on computing the square root [47, 48, 45]. We introduced an interval method for square root computation, accompanied by two significant results. Firstly, we extended state-of-the-art work by synthesizing a faster converging refinement map using quantifier elimination techniques. Secondly, we developed efficient simplification techniques for sign semi-definite conditions, showing their effectiveness compared to existing tools. This research is the continuation from my PhD thesis. I appreciate my contribution on [47, 48] is 70% in terms of formalization, related work, methodology, experimental results, writing and revision. The contribution on [45] is 100% but was inspired by previous works on which I was co-author.

The contributions from Section 3.2 are around the exploration of security and provisioning challenges in Cloud computing, encompassing contributions that: *(1)* establish a service catalog for security capabilities, *(2)* introduce an enhanced Security SLA model and provisioning framework, and *(3)* address the intricate task of automated deployment for component-based applications in the Cloud. For the first two papers [32, 31], I appreciate my contribution is 20% in terms of methodology, writing, revisions. For the third contribution, I appreciate my contribution 40% on [52], while, for the other papers, co-authored with my students I appreciate my contribution to 50% in terms of formalization, related work, methodology, solution, writing, revision.

The contributions from Section 3.3 highlight the significance of Big Data in supporting data-intensive applications (DIAs). The long-term goal is to represent Apache Storm topologies' runtime behavior at design-time. Three key contributions are presented: *(1)* An automated verification approach using CLTLoc temporal logic to check queue growth in Storm topologies. I appreciate my contribution on [78] is 20% in terms of formalization, related work, writing and revision. *(2)* Formalization and automation of verification for data-intensive applications using the array-based systems approach with the Cubicle model checker. I appreciate my contribution on [20] is 40% in terms of formalization, related work, methodology, experimental results, writing and revision. *(3)* Introducing the DICE Verification Tool (D-VerT) for assessing designs and safety properties, available as an open-source resource on GitHub. These advancements facilitate efficient and robust analysis of Storm applications, reducing the need for post-deployment redesigns. I appreciate my contribution on [19] is 10% in terms of formalization, methodology, writing and revision.

The contributions from Section 3.4 are showcased in two key areas: fake news detection and traffic sign recognition. The first contribution involves a comprehensive tool for fake news detection, featuring a problem statement, innovative methodology including parsing and machine learning, and thorough implementation details.

Experimental results validate the tool's effectiveness, particularly in content and title detection accuracy. In the second contribution, the candidate explores binarized neural networks (BNNs) for traffic sign recognition, presenting an in-depth methodology that integrates XNOR architectures and internal blocks. With well-defined datasets and experimental settings, the candidate's expertise in this field is evident, further validated by practical outcomes presented in the experimental results. I appreciate my work on [10, 91], co-authored with my students, is 50% in terms of formalization, related work, methodology, solution, writing, revision.

The contributions from Section 3.5 shed light on the impact of transferring learning into the workplace for an Informatics teacher (the thesis writer) promoting student-centered learning (SCL) in Software Engineering. Employing a quasi-experimental design, the study reveals positive effects on student learning approaches and teaching quality as perceived by students. I appreciate my contribution on [53] is 50% in terms of formalization, related work, methodology, experimental results, writing and revision.

Chapter 2

Scientific and Professional Career: Past and Present

2.1 Important scientific achievements of the candidate

I have started doing research in *Symbolic Computation* and *Formal Methods* since my Master and PhD studies at Research Institute for Symbolic Computation[1], Linz, Austria, one of the top institutions worldwide, doing research in *automated reasoning* and *symbolic computation.* I was working in the Theorema group[2], led by the inventor of Groebner Basis, Bruno Buchberger. During my PhD studies I was researching on techniques of computational logic and quantifier elimination by cylindrical algebraic decomposition for (semi-) automatic static analysis and synthesis of algorithms [44]. I was supervised by and co-authored papers with leading Symbolic Computation scientists: Tudor Jebelean[3] and Hoon Hong[4] (former editor-in-chief of the Journal of Symbolic Computation[5]). Some of our results, here mentioned those published after PhD graduation, appeared in the most important symbolic computation venue (ISSAC http://www.issac-conference.org – category A*) and journal (J. of Symbolic Computation – category Q2). Our works are acknowledged by the community being cited in the prestigious venues and journal: ISSAC and Journal of Symbolic Computation.

Once I returned in Romania, in 2014, my research agenda mainly focused on *making formal methods amenable to being applied to hard problems of AI*, like resource management in the Cloud or verification of Big-Data applications and, more recently, machine learning. This was facilitated by participating in various EU projects where I worked on international teams with senior or junior researchers. In the Secure Provisioning of Cloud Services based on SLA management (SPECS)[6] project, I was involved in the resource management in the Cloud problem which is a hard-combinatorial optimization problem. The outcome of our work was published in IEEE Trans. on Services Computing [31] (category Q1) but also at confer-

[1] `https://risc.jku.at`

[2] `http://www3.risc.jku.at/research/theorema/software/`

[3] `https://www3.risc.jku.at/people/tjebelea/`

[4] `https://hong.math.ncsu.edu`

[5] `https://www.sciencedirect.com/journal/journal-of-symbolic-computation`

[6] `https://cordis.europa.eu/project/id/610795`

ences [32]. This line of research inspired the project MANeUveR (MANagement agency for cloUd Resources)[7] and also the current research project that I am coordinating: SAGE (A Symbiosis of Satisfiability Checking, Graph Neural Networks and Symbolic Computation)[8].

In another EU project, Developing Data-Intensive Cloud Applications with Iterative Quality Enhancements (DICE)[9], I was working on the verification of data-intensive applications based on the big-data technology, Apache Storm[10]. This kind of applications are difficult to be verified since they are abstracted as infinite-state systems. For verification purposes, we have used both first-order logic formalisms (array-based systems), as well as metric temporal logic (CLTLoc) and managed to check safety and liveness properties of the systems. In this project, I have strengthened and gained expertise in advanced logical formalisms used in formal methods. The results are published at important venues for the formal methods community [19, 20] and cited in, e.g. ACM Computing Surveys, Journal of Grid Computing, Euromicro Conference on Software Engineering and Advanced Applications.

The first national project led by me, MANeUveR, was an experimental demonstration project (PED) ended on December 2018. In the framework of MANeUveR, we studied the resource provisioning in the Cloud problem [52] consisting in the allocation of virtual machines (VMs) from various Cloud Providers (CPs) to a set of applications such that the constraints induced by the interactions between components and by the components hardware/software requirements are satisfied and the performance objectives are optimized (e.g. costs are minimized). We formulated it as a constrained optimization problem and tackled by exact (combined constraint programming and optimization modulo theory with symmetry breaking techniques for state-space reduction) [52, 51], approximate methods (population-based metaheuristics) [82]. The most important outcomes of MANeUveR is [52] (category Q2) as well as a prototype implementation of a recommendation engine[11]. Our work is cited in, e.g. Journal of Systems and Software, Computers & Security, ACM Symposiom on Applied Computing.

In the current project I am coordinating, SAGE, we refine and combine methods from satisfiability checking, graph neural networks and symbolic computation for the analysis and synthesis of complex systems. More precisely, one research line is in the direction of the robustness of machine learning classifiers for traffic sign recognition. Traffic signs are crucial for road safety and autonomous driving, hence, in [91] motivated by: *(1)* convolutional neural networks (CNNs) are widely used for traffic sign recognition, *(2)* existing CNNs architectures have an accuracy of 99.46% however the models are large and unsuitable for low consumption devices, and *(3)* binarized neural networks (BNNs) are highly unexplored although offer compact architectures with accuracy similar to CNNs, we proposed BNN architectures achieving over 90% for German Traffic Sign Recognition Benchmark and an average above 80% across other datasets, with relatively medium number of parameters. The best accuracy models trained were sent to this year competition

[7] https://merascu.github.io/links/MANeUveR.html

[8] https://merascu.github.io/links/SAGE.html

[9] https://cordis.europa.eu/project/id/644869

[10] https://storm.apache.org

[11] https://github.com/Maneuver-PED

on verification of neural networks (VNN-COMP'23) and considered fairly complex for state-of-the-art tools[12].

Another line of research of SAGE is inspired by the Cloud deployment problem studied in MANeUveR. In SAGE our efforts are towards: *(a)* providing an understanding of what the symmetries and similarities are for the resource management in the Cloud *(b)* developing methods for breaking the symmetries in and for learning templates from the problem *(c)* inventing theory and algorithms for abstracting away the symmetries of these case studies and for capturing the commonalities of the symmetry breaking techniques using the theory of invariant groups and SAT/SMT solving, *(d)* learning problem templates by formalizing the problem as a GNN and applying on-the-shelf GNNs libraries for different predictions *(e)* studying the computational effectiveness of the newly developed symmetry breaking and similarity breaking techniques performing automatic solving of constraint satisfaction problems (CSP) at scale. Some results obtained so far are under review.

My researcher profiles on certain platforms are as follows:

- ISI Web of Science: `http://apps.webofknowledge.com/Search.do?product=UA&SID=P1aYgkPSh7xEVK1yjqP&search_mode=GeneralSearch&prID=c308286d-37f9-44dd-9943-7dee4e81b212`
- Scopus: `http://www.scopus.com/authid/detail.uri?authorId=36093715900`
- ACM Digital Library: `http://dl.acm.org/author_page.cfm?id=81461658454&CFID=560435087&CFTOKEN=31590081#`
- Google Scholar: `https://scholar.google.ro/citations?user=2Dbzyq8AAAAJ&hl=ro`
- DBPL: `http://dblp.uni-trier.de/pers/hd/e/Erascu:Madalina`

A detailed description of the scientific achievements agaist state-of-the-art is presented in Chapter 3.

2.2 Fundraising and coordination of research teams

Regarding fundraising and abilities to coordinate research teams, I was/am the PI of:

- Sept 2022 - Aug 2024: SAGE: A Symbiosis of Satisfiability Checking, Graph Neural Networks and Symbolic Computation. Funding agency: UEFISCDI. Budget: 416000 RON (approx. 93000 EUR). Competitive competition: 8 financed projects out of 37 submitted. Website: https://merascu.github.io/links/SAGE.html. Refereed publication so far: [91]. For an exhaustive list of achievements and activities, one could check the homepage of the project.

[12] `https://sites.google.com/view/vnn2023`

- Sept 2017 – Dec 2018: *MANeUveR: Management Agency for Cloud Resources*. Funding agency: UEFISCDI. Budget: 475000 RON (approx. 102000 EUR). Competitive competition: 168 financed projects out of 2074 submitted. Website: https://merascu.github.io/links/MANeUveR.html. Publications: [52, 82, 51, 49, 50]. We also obtained a mention award at BringITon2018[13], Iasi, Romania, a workshop for promoting and capitalizing the interaction between computer science in academia and business environment.

- I was also responsible for the Institute eAustria[14] team for different WPs in SPECS and DICE projects described above.

2.3 Facilitating research among students

I also advised students on research topics which were accepted at student symposia [71], workshops [50, 38], conferences [10, 91]. The most talented ones were part of the team of the research projects MANeUveR and SAGE.

2.4 Other research related achievements

Another proof that my work is acknowledged by the communities where I am active in is my participation as an associated member in the project SC-square: Satisfiability Checking and Symbolic Computation[15] ended in August 2018. The project was funded by the EU H2020 research and innovation program under FET Open call known for supporting radically new future technologies. The project successfully initiated cross-fertilization of Satisfiability Checking and Symbolic Computation fields and brought mutual benefits. It enabled the development of improved software tools by combining the knowledge, experience, and the technologies of these rather distinct communities. A continuation of this project is envisioned and I was invited to a preliminary brainstorming group for New Perspectives in Symbolic Computation and Satisfiability Checking[16].

Currently, I am part of the COST action European Research Network on Formal Proofs[17] and I serve as the leader of the Working Group 3 Program Verification. This is a position not remunerated, but voluntary, with the aim of fostering collaboration between program verification researchers, and not only, in Europe. Some activities include the organization of the annual meeting [18], coordination and contribution to the deliverables, planning of the 2024 meetings of the working group organized by other members of the action, participation in brainstorming seminars on future persoectives on formal proofs [19].

I am also active in the Romanian formal methods community being an invited speaker at Working Formal Methods Symposium (FROM 2019)[20] or giving a con-

[13] http://bringiton.info.uaic.ro

[14] https://ieat.ro

[15] http://www.sc-square.org/people.html

[16] https://www.dagstuhl.de/en/program/calendar/semhp/?semnr=22072

[17] https://europroofnet.github.io

[18] https://europroofnet.github.io/wg3-timisoara/

[19] https://www.dagstuhl.de/en/seminars/seminar-calendar/seminar-details/23401

[20] https://from2019.projects.uvt.ro

tributed talk at Theoretical Computer Science, Operations Research and Optimization part of The Tenth Congress of Romanian Mathematicians 2022[21].

I was the Chair of the Doctoral Program of The 15th Conference on Intelligent Computer Mathematics (CICM 2022), September 19 - 23, 2022, Tbilisi, Georgia. I served as PC member of various conferences, e.g. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), International Conference on Artificial Neural Networks (ICANN), International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), International SPIN Symposium on Model Checking of Software (SPIN). I also organized The 14th Conference on Intelligent Computer Mathematics (CICM 2021).

I also served as a referee for Funding Agencies, research calls: European Commission (HORIZON-SESAR-2022 call); University of Parma: Program for research projects submitted by internal junior researchers; Italian Ministry for University and Research (MUR), PRIN call.

2.5 Teaching activities

At West University, starting 2014, I engaged in teaching activities (seminars, laboratories) in both Romanian and English: Computational Logic, Algorithmics, Data Structures, Programming I (C programming language), Programming III (Java programming language) – at Bachelor level; Basic Techniques in Scientific Activity, Automated Theorem Proving – at Master level.

In 2016, I became a Lecturer, then, since 2021, I am Associate Professor in the Department of Computer Science, teaching predominantly at the undergraduate level, subjects such as Software Engineering, Formal Methods in Software Development (in both Romanian and English specializations), Formal Languages and Automata Theory (both lecture and seminar/laboratory format), Methodology and BSc Thesis Preparation. Additionally, I have taught Master's level courses such as Special Topics in Artificial Intelligence, Formal Verification. In addition to these activities, I have supervised the internship programs for undergraduate students.

I have continuously tried to improve and innovate my teaching activities. First, by exposing the students to research earlier in their studies by introducing the research I am interested in my lectures, and, second, by adopting a student-centered teaching, learning and evaluation. In February 2020, I attended a course offered by the Academic Development Center of West University, focused on University Didactics and Psychopedagogy, specifically Curriculum Design, Modern Teaching Methods, Development of Educational Materials, and Student Group Management. The purpose of this course was to present the reflective-collaborative teaching and assessment model, which places a strong emphasis on student-centered learning. This model proved to be highly valuable, especially in the context of the Covid19 pandemic as reported in the paper [53]. However, my personal conclusion is that students are not fully prepared to adopt this approach, as they are more accustomed to traditional exams during exam sessions rather than ongoing assessment with problem-solving scenarios for analysis. Moreover, for constant feedback for students' assignment, more specialized human resources are needed which is not possible due to the fact that the education is underfunded thus unattractive.

[21] http://www.imar.ro/~congmatro10/sections.html

I am also interested in Innovation and Entrepreneurship in Technology and Education Recently, I was awarded the Fulbright-RAF award in the domain of Entrepreneurship and Entrepreneurial Studies for a semester-long fellowship (Spring 2022) in residence at University of Rochester, AIN Center for Entrepreneurship, USA. The expertise gained started being used in the framework of lectures on entrepreneurship skills for Computer Science Students (mandatory discipline at Bachelor level) but not only (complementary/transversal discipline for all university students at Bachelor level).

Chapter 3

Presentation of the Research Contributions against State-of-the-art

3.1 Formal Methods in Symbolic Computation

This section presents the investigations of applying real quantifier elimination to synthesize optimal numerical algorithms. As a case study, we focus on computing the square root of a given real number which is a fundamental operation. Naturally, various numerical methods have been developed [54, 86, 107, 80, 84, 65, 36, 6, 17, 94, 85]. We consider an interval version of the problem [84, 6, 85]: given a real number x and an error bound ε, find an interval such that it contains $\sqrt{x}$ and its width is less than ε. One way to solve the problem starts with an initial interval and repeatedly updates it by applying a *refinement* map, say R, on it until it becomes narrow enough (see Algorithm 1).

Algorithm 1 Interval method for square root

```
Require: $x > 0, \ \varepsilon > 0$
Ensure: $I$, an interval such that $\sqrt{x} \in I$ and $\mathsf{width}(I) \le \varepsilon$
  $I \leftarrow [\min(1, x), \max(1, x)]$
  while $\mathsf{width}(I) > \varepsilon$ do
      $I \leftarrow R(I, x)$
  return $I$
```

We present two results.

1. We extend the result of [46] by synthesizing a new *refinement map* which converges faster than the well-know Secant-Newton. First, we consider the general class of refinement maps which includes also Secant-Newton. Then we impose that the class of maps are correct, terminating and optimal. Additionally, we imposed a mild/natural constraint in order to reduce the search space. Further, we transform the synthesis problem into a problem of quantifier elimination over real numbers. Since the quantifier elimination problem is computationally intractable for general purpose quantifier elimination tools, we had to:

- carefully reduce a complicated quantified formula into several simpler ones, and
- automatically eliminate the quantifiers from the resulting ones using the state-of-the-art quantifier elimination software.

Consequently, we synthesized the optimal refinement map:

$$
\begin{aligned}
L' &= L + \frac{x - L^2}{L + U} \\
U' &= U + \frac{x - \frac{3}{4}U^2 - \frac{1}{2}LU + \frac{1}{4}L^2}{U + L}
\end{aligned}
$$

This result appeared in [47], was extended in [48] and it is presented in Section 3.1.1.

2. By abstracting the hand derivations used for the simplifications of the quantifier elimination problem in [47, 48], we came up with efficient simplification techniques tailored for sign semi-definite conditions (SsDCs). The SsDCs for a polynomial $f \in \mathbb{R}[y]$ with parametric coefficients are written as $\underset{\substack{y \\ L \le y \le U}}{\forall} f(y) \geq 0$ and $\underset{\substack{y \\ L \le y \le U}}{\forall} f(y) \leq 0$. We give sufficient conditions for the simplification techniques to be sound for linear and quadratic polynomials. We show their effectiveness compared to state of the art quantifier elimination tools for input formulae occurring in the optimal numerical algorithms synthesis problem by an implementation on top of `Reduce` command of Mathematica. Additionally, we proved that the results from the previous item hold if one removes the termination condition (because we know there exists at least a terminating one) and the constraint imposed for shrinking the search space. This result appeared in [45] and it is presented in Section 3.1.2.

3.1.1 Real quantifier elimination for the synthesis of optimal numerical algorithms (Case study: Square root computation)

This section is about the application of real quantifier elimination to the synthesis of optimal numerical algorithms. The motivation is represented by the hand-crafted refinement map (called *Secant-Newton*) which is generalized in order to find better refinement maps.

Secant-Newton map is obtained by combining the secant map and the Newton map where the secant/Newton map is used for determining the lower/upper bound of the refined interval, that is,

$$
\begin{aligned}
R :&\ [L, U], x \mapsto [L', U'] \\
L' &= L + \frac{x - L^2}{L + U} \\
U' &= U + \frac{x - U^2}{2U}
\end{aligned}
$$

which can be derived from Figure 3.1.[1] [2] A question naturally arises. *Is there any refinement map which is better than Secant-Newton?* In order to answer the question rigorously, we applied the following methodology:

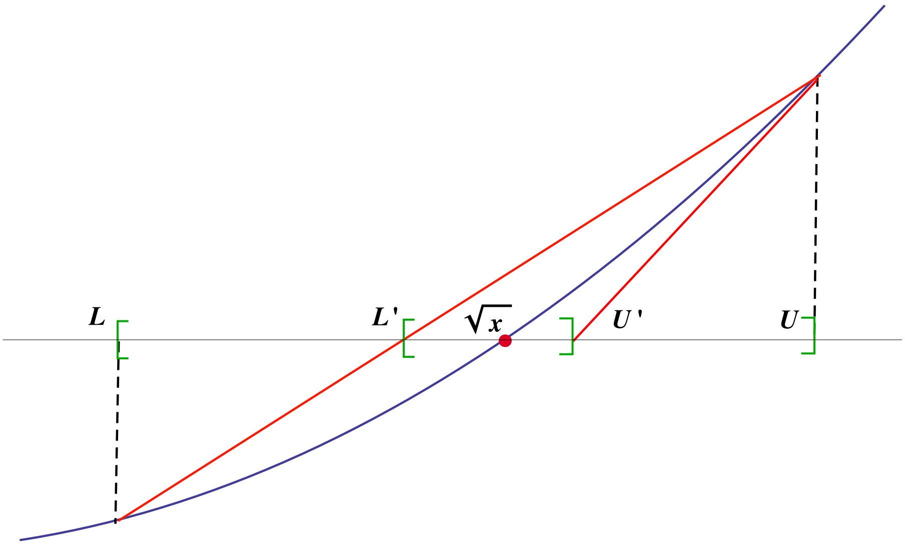


Figure 3.1: Derivation of Secant-Newton map

1. We fixed a search space, that is, a family of maps in which we search for a better map. This was done by observing that the Secant-Newton map "scales properly", that is, if we multiply $\sqrt{x}$, L and U by a number, say s, then L' and U' are also multiplied by s. This is due to the fact that the numerators are quadratic forms in $\sqrt{x}$, L and U and the denominators are linear forms. This observation suggests the following choice of a search space: the family of all the maps with the form

$$R_{p,q}: \ [L,U], x \mapsto [L',U']$$
$$L' = L + \frac{x + p_0L^2 + p_1LU + p_2U^2}{p_3L + p_4U} \qquad U' = U + \frac{x + q_0U^2 + q_1UL + q_2L^2}{q_3U + q_4L}$$

   which we call *quadratic maps*[3]. By choosing the values for the parameters $p = (p_0, \ldots, p_4)$ and $q = (q_0, \ldots, q_4)$, we obtain each member of the family. For

[1] An anonymous referee made an interesting observation that the Secant-Newton map can be also viewed as an instance of the interval Newton map with slope:

$$[L,U], x \quad \mapsto \quad m - \frac{m^2 - x}{m + [L,U]}$$

where $m \in [L,U]$. If we choose $m = U$ then it is identical to the Secant-Newton map.

[2] It is important to note that there are faster non-interval algorithms for computing square roots [80, 65, 36, 17]. They are based on static error analysis, auto-corrective behavior of Newton map, etc. However, in this paper, we restrict our investigation to interval methods because the current work is carried out as a preliminary study, in the hope of identifying conceptual and technical tools for finding an optimal method for solving polynomial equations. Interval based methods have the benefit of providing a uniform paradigm for such larger class of problems.

[3] A careful reader would be concerned about the possibility of the denominators becoming 0, making the expressions undefined. Fortunately it will turn out that these cases will be naturally eliminated in the subsequent discussions.

instance, Secant-Newton map can be obtained by setting $p = (-1, 0, 0, 1, 1)$ and $q = (-1, 0, 0, 2, 0)$.

2. We imposed the conditions that these maps are correct but no necessarily contracting $C(p, q)$[4], terminating $T(p, q)$, and quadratic convergent $Q(p, q)$. Additionally, we imposed the minimization of the so-called Lipschitz constant $E(p, q)$, which measures the complexity of the algorithm (the smaller, the faster).

$$
\begin{aligned}
C(p,q) \quad &:\Longleftrightarrow \quad \underset{\substack{L,U,x\\0<L\leq\sqrt{x}\leq U}}{\forall} \quad 0 < L' \leq \sqrt{x} \leq U' \\
T(p,q) \quad &:\Longleftrightarrow \quad \underset{\substack{x\\x>0}}{\forall} \;\; \underset{\substack{c\\1>c>0}}{\exists} \;\; \underset{\substack{L,U\\0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L) \\
Q(p,q) \quad &:\Longleftrightarrow \quad \underset{\substack{x\\x>0}}{\forall} \;\; \underset{\substack{c\\c>0}}{\exists} \;\; \underset{\substack{L,U\\0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L)^2 \\
E(p,q) :&= \sup_{\substack{0<L\leq\sqrt{x}\leq U\\L\neq U}} \tfrac{U'-L'}{U-L}
\end{aligned}
$$

3. We formulated the constrained optimization problem:

$$\text{Minimize } E(p,q) \text{ subject to } C(p,q) \wedge T(p,q) \wedge Q(p,q)$$

4. We applied state-of-the-art quantifier elimination tools to solve the optimization problem above, however it was computationally intractable, hence:
   (a) we reduced a complicated quantifier elimination formula into several simpler ones, by:
      i. carefully dividing the formula, exploiting the logical structure of the formula,
      ii. judiciously instantiating some of quantified variables, exploiting the algebraic structure of the formula,
      iii. imposing a mild/natural restriction on the search space for p, q which tremendously speeded up the process of quantifier elimination; this restriction is obtained by adding one more constraint

      $$K(p,q) \;:\Longleftrightarrow\; 0 < p_4 \leq 2 \,\wedge\, p_3 + p_4 = 2 \,\wedge\, 0 < q_3 \leq 2 \,\wedge\, q_3 + q_4 = 2$$

      in addition to $C(p, q)$, $T(p, q)$ and $Q(p, q)$.
      The condition K was motivated by the wish that the critical points of some functions appearing in the quantifier elimination of the condition C lie in the interval $[L, U]$. We conjecture that the condition K can be dropped without changing the main theorem. We leave it as an open challenge.

   (b) we eliminated the quantifiers resulting several simpler formulas automatically, using the state-of-the-art quantifier elimination software such as Mathematica [108] and QEPCAD-B [28].

Remark 3.1. For solving the constrained optimization problem, one cannot simply apply standard numerical optimization methods to the above optimization problem, due to the following reasons:

[4] We proved in [46] that Secant-Newton is optimal among contracting maps.

1. The constraints C, T and Q are quantified formulas.
2. The objective function E is the result of parametric optimization (sup).
3. It turns out that there are infinitely many values of p and q with the same minimum Lipschitz constant.

Hence, the problem was translated into a real quantifier elimination problem.

As the result, we were able to synthesize semi-automatically an optimal quadratically convergent map, which is better than the well known hand-crafted Secant-Newton map. Interestingly, the optimal synthesized map is not contracting as one would naturally expect.

In this chapter we focus on the results obtained in [48] as they generalize those obtained in [47], more precisely:

1. In [47], we optimized among the correct and terminating maps, obtaining an infinite family of optimal maps. In this paper, we optimized among the correct, terminating and *quadratically convergent* maps, obtaining an unique optimal map. It was ensured by adding the following constraint

$$Q(p,q) \quad :\Longleftrightarrow \quad \underset{\substack{x \\ x>0}}{\forall} \;\; \underset{\substack{c \\ c>0}}{\exists} \;\; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U'-L' \leq c\,(U-L)^2$$

   This solves one of the open problems posed in [47].

2. In [47], we formulated the termination constraint T as follows.

$$T(p,q) \quad :\Longleftrightarrow \quad \underset{\substack{c \\ 1>c>0}}{\exists} \;\; \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U'-L' \leq c\,(U-L)$$

   In this paper, we changed it to the following formulation

$$T(p,q) \quad :\Longleftrightarrow \quad \underset{\substack{x \\ x>0}}{\forall} \;\; \underset{\substack{c \\ 1>c>0}}{\exists} \;\; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U'-L' \leq c\,(U-L)$$

   because it reflects the termination constraint more naturally: the value of the constant c may depend on the input x.

Main Result

In this section we state the main result. The details of the proof can be checked in [45].

Theorem 3.1 (Main)**.** *We have*

(A) $C \wedge T \wedge Q \wedge K \quad \Longrightarrow \quad E \geq \frac{1}{4}$

(B) $C \wedge T \wedge Q \wedge K \wedge E = \frac{1}{4} \quad \Longleftrightarrow \quad p = (-1,0,0,1,1) \;\wedge\; q = (-\frac{3}{4}, -\frac{1}{2}, \frac{1}{4}, 1, 1)$

Remark 3.2. Theorem 3.1 essentially states that the optimal quadratic convergent map has the following form

$$\bar{R}(I,x) = \left[L + \frac{x-L^2}{L+U}, U + \frac{x - \frac{3}{4}U^2 - \frac{1}{2}LU + \frac{1}{4}L^2}{U+L}\right]$$

and that its Lipschitz constant is $\frac{1}{4}$.

Remark 3.3. The Secant-Newton map is also quadratic convergent and has the Lipschitz constant $\frac{1}{2}$ (see Mathematica).

Remark 3.4. Theorem 3.1 is a curious result in the light of the previous finding [46] that Secant-Newton is the optimal among the "contracting" quadratic maps, that is, among the quadratic maps satisfying the strong condition

$$\underset{\substack{L,U,x \\ 0<L\le\sqrt{x}\le U}}{\forall} \quad 0 < L \le L' \le \sqrt{x} \le U' \le U.$$

In the present paper, we dropped the requirement of contracting, allowing $L' < L$ or $U' > U$. In fact, the synthesized map is only left contracting (see Mathematica), that is, it satisfies

$$\underset{\substack{L,U,x \\ 0<L\le\sqrt{x}\le U}}{\forall} \quad 0 < L \le L' \le \sqrt{x} \le U.$$

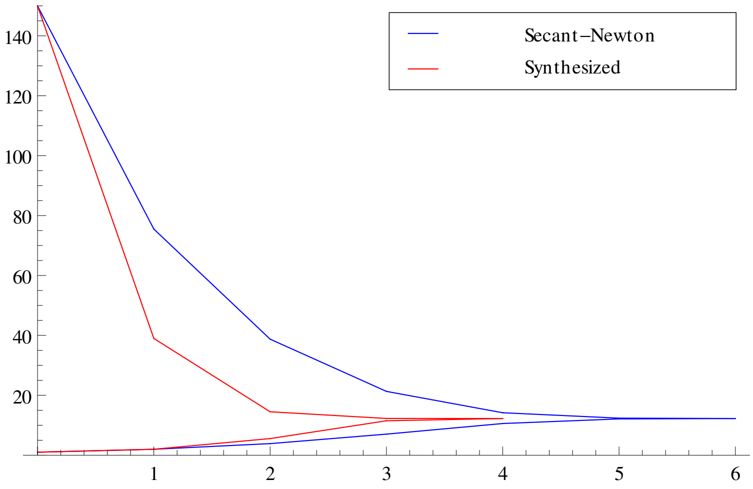
It is very counter-intuitive that such a non-contracting map could be better than contracting ones. It might explain why such a optimal map was not discovered, through hand-crafting, until now.

Remark 3.5. Let Secant-Newton map be R^* and the newly synthesized map $\bar{R}$:

$$\begin{aligned} R^*(I,x) &= \left[L + \frac{x-L^2}{L+U}, U + \frac{x-U^2}{2U}\right] &&= \left[\frac{x+LU}{L+U}, \frac{x}{2U} + \frac{2U}{4}\right] \\ \bar{R}(I,x) &= \left[L + \frac{x-L^2}{L+U}, U + \frac{x - \frac{3}{4}U^2 - \frac{1}{2}LU + \frac{1}{4}L^2}{U+L}\right] &&= \left[\frac{x+LU}{L+U}, \frac{x}{L+U} + \frac{L+U}{4}\right] \end{aligned}$$

The above rewriting of R^* and $\bar{R}$ reveals a small but crucial difference: $2U$ in R^* is replaced with $L+U$ in $\bar{R}$. This small change reduces the Lipschitz constant by half. It also shows that the number of arithmetic operations needed for one application of R^* and that for $\bar{R}$ are the same.

Remark 3.6. We ran the Algorithm 1 using the Secant-Newton map R^* and the synthesize map $\bar{R}$ on the input $x = 150$ and $\varepsilon = 10^{-5}$. The trace of intermediate results is shown bellow. The horizontal axis represents the iteration number and the vertical axis represents the L and U values.



Remark 3.7. The fact that the Lipschitz constant of the synthesized map is smaller than that of Secant-Newton map does *not* imply that for all the values of L, U, x the synthesized map is better than the Secant-Newton map, since the Lipschitz constant is about the worst case situation. Therefore, a question arises naturally: *When is the Secant-Newton map better than the synthesized map?* It is easy show that this happens iff the condition $U\frac{L+U}{2} < x$ holds, which roughly means that $\sqrt{x}$ is "sufficiently" close to U.

Hence we have another natural question: *Does this situation actually arise during the execution of the synthesized algorithm (Algorithm 1 using the synthesized map $\bar{R}$)?* The answer is *Yes, it does.* As an example, consider the input $x = 4$ and $\varepsilon = 2^{-4}$. Initially we have $[L, U] = [1, 4]$. After the first iteration, we have $[L, U] = [\frac{8}{5}, \frac{41}{20}]$. One can easily verify that the condition $U\frac{L+U}{2} < x$ is satisfied.

Curiously, for the above particular input, in spite of the fact that the condition $U\frac{L+U}{2} < x$ holds right after the first iteration, one can easily verify that the synthesized algorithm is still better than the Secant-Newton algorithm in that the synthesized algorithm requires 2 iterations while the Secant-Newton algorithm requires 3 iterations.

Thus it leads us to the ultimate question. *Is the Secant-Newton algorithm ever better than the synthesized algorithm?* In order to get an initial rough answer, we ran both algorithms on various values of inputs x and ε, collecting some relevant data. The implementation of both algorithms is done on Maple and can be found in:

`http://www.risc.jku.at/projects/SPy/JSC2015/MapleFiles/JSC2015.mw`.

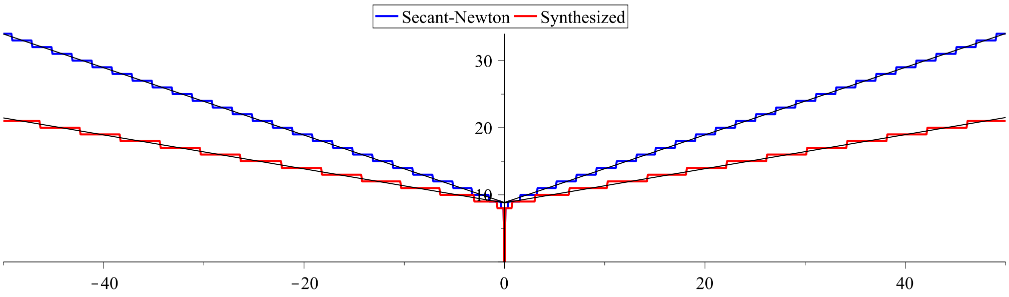


Figure 3.2: # of iterations of the Secant-Newton algorithm and the synthesized algorithm

In Figure 3.2, the horizontal axis represents $\log_2(x)$ and the vertical axis represents the number of iterations required to compute $\sqrt{x}$ with precision $\varepsilon = 2^{-1000}$. We have tried various different values for ε and found that all the resulting graphs have a very similar shape as the one given here. From the figure, one could conjecture that the Secant-Newton algorithm is never better than the synthesized algorithm. Proving (or disproving) the conjecture is left open as a future work.

By fitting the above graphs to lines, we see that the slopes of the blue lines (Secant-Newton) and the red lines (Synthesized) are roughly 0.5 and 0.25 respectively, which are the reciprocals of the Lipschitz constants. This suggests that the Lipschitz constant turns out to be useful for studying the actual performance of the algorithms even though it is defined to capture only the worst possible performance.

Proof (Synthesis)

In this section, we state the main result (Theorem 3.1) and present the proof steps. The detailed proof can be checked in the paper [48].

The proof steps are as follows.

1. Eliminate quantifiers from the constraint $C \wedge T \wedge Q \wedge K$, obtaining an equivalent quantifier-free condition, say F.

2. Eliminate sup from the objective function E, that is, carry out parametric maximization of E, obtaining an expression, say G.

3. Minimize G subject to F.

In principle, all these could be carried out using state-of-the-art real quantifier elimination software (e.g. Reduce command in Mathematica [108], QEPCAD-B [28], and Redlog [43]) and symbolic optimization software (e.g. Maximize/Minimize command in Mathematica [108]).

However, due to huge computational requirement, they could not be carried out in practice. Thus, we proceeded as follows:

1. *Simplify the constraint*: We found a suitable quantifier-free necessary condition to the constraint $C \wedge T \wedge K$ (without Q) by (1) dividing the formulas into simpler ones, (2) reducing the number of quantified variables by judicially instantiating some of them, and then (3) eliminating remaining quantifiers using Mathematica and QEPCAD-B.

2. *Simplify the objective function*: We eliminated the sup from the objective function E by (1) carrying the maximization over a few variables by hand, and then (2) carrying the maximization over the remaining variable using Mathematica.

3. *Carry out the constrained minimization* : We carried out the minimization of the simplified objective function under the simplified constraint, using Mathematica, obtaining an infinite family of optimal maps. Choose among them the ones that satisfy $C \wedge T \wedge Q \wedge K$.

We used Mathematica and QEPCAD-B for all computations. Redlog could have been used also. However we have not used it, mainly because it delivered unsimplified output. By default, we used Mathematica because it incorporates both quantifier elimination and constraint optimization routines. We used QEPCAD-B when Mathematica output is not simplified enough.

The complete logs of the inputs and the outputs of those programs can be found in the following link:

`http://www.risc.jku.at/projects/SPy/JSC2015/`

3.1.2 Efficient Simplification Techniques for Special Real Quantifier Elimination with Applications to the Synthesis of Optimal Numerical Algorithms

This section presents:

- simplification techniques motivated by our previous work [47, 48]. More precisely, in the quantifier elimination (QE) from the constraint C, which ensures the correctness of the Algorithm 1, high degree of manual intervention was needed since the general QE tools failed to solve the QE problem. Hence, by abstracting these hands-on techniques into systematic ideas, we devised simplification methods which assist the general QE by CAD algorithm from Mathematica (`Reduce` command) in delivering a quantifier-free formula.

- extensions of the main result (Theorem 3.1) from [47, 48] by dropping out the termination condition T and the assumption K. The result obtained previously does not change. However, the solution process is similar to the one performed in [47, 48].

The simplification/preprocessing methods are for the following families of formulae:

$$\underset{\substack{y\\L\leq y\leq U}}{\forall}\ f(y)\geq 0 \qquad\qquad \underset{\substack{y\\L\leq y\leq U}}{\forall}\ f(y)\leq 0$$

where $f\in\mathbb{R}[y]$, $f(y)=y^n+a_{n-1}y^{n-1}+...+a_1y+a_0$ where $a_{n-1},...,a_0$ are real parameters. More precisely, we give quantifier-free conditions equivalent to each of them. These conditions depend on the behavior (monotone, convex, concave) of f on the interval $[L,U]$, hence we give sufficient conditions for a certain behavior for linear and quadratic case.

Extensions (Theorem 3.1) of the main result refer to:

1. In [47, 48], we optimized also among the terminating maps, that is among the maps satisfying the condition:

   $$T(p,q)\quad:\Longleftrightarrow\quad \underset{\substack{x\\x>0}}{\forall}\ \ \underset{\substack{c\\1>c>0}}{\exists}\ \ \underset{\substack{L,U\\0<L\leq\sqrt{x}\leq U}}{\forall}\quad U'-L'\leq c\,(U-L)$$

   Since we know that we have at least one map for which $c\in(0,1)$ (Secant-Newton map has $c=\frac{1}{2}$) the condition T is left out.

2. In [47], we optimized among the correct and terminating maps which fulfilled a certain natural condition, obtaining an infinite family of optimal maps. In [48], we detected in the class of optimal maps *a single optimal quadratically convergent map.* In this paper, we removed the natural condition, enlarging the class of optimal maps of [47] while the main result of [48] is preserved. The removal of the additional constraint solves one of the open problems posed in [47, 48].

Simplification Techniques for Sign Semi-Definite Conditions

In this section we introduce the definitions of sign semi-definite conditions, simplification techniques tailored for them, as well as sufficient conditions for the simplification techniques to be sound.

Definition 3.1. Let $f(y) = y^n + a_{n-1}y^{n-1} + ... + a_1 y + a_0$ be a polynomial in y over $\mathbb{R}$. We call the following conditions sign semi-definite conditions (SsDCs)[5] for f:

$$\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \ge 0 \qquad\qquad \underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \le 0$$

Definition 3.2. f is monotone increasing on $[L, U]$ iff $\underset{L \le x \le y \le U}{\underset{x,y}{\forall}} f(L) \le f(x) \le f(y) \le f(U)$.

The following lemmas (Lemmas 3.2 - 3.5) eliminate one universally quantified variable from an univariate polynomial expression providing equivalent necessary and sufficient conditions. Note, however, that they can be applied also to multivariate polynomials by viewing the multivariate polynomials as univariate polynomials.

Lemma 3.2. *Let f be monotone increasing on $[L, U]$. Then we have*

(a) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \ge 0 \iff (L \le U \Rightarrow f(L) \ge 0)$

(b) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \le 0 \iff (L \le U \Rightarrow f(U) \le 0)$

Lemma 3.3. *Let f be monotone decreasing on $[L, U]$. Then we have*

(a) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \ge 0 \iff (L \le U \Rightarrow f(U) \ge 0)$

(b) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \le 0 \iff (L \le U \Rightarrow f(L) \le 0)$

Lemma 3.4. *Let f be convex on $[L, U]$. Then we have*

(a) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \ge 0 \iff (L \le c \le U \Rightarrow f(c) \ge 0)$*, where c is the critical point of f.*

(b) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \le 0 \iff (L \le U \Rightarrow f(L) \le 0 \wedge f(U) \le 0)$

Lemma 3.5. *Let f be concave on $[L, U]$. Then we have*

(a) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \ge 0 \iff (L \le U \Rightarrow f(L) \ge 0 \wedge f(U) \ge 0).$

(b) $\underset{L \le y \le U}{\underset{y}{\forall}} \; f(y) \le 0 \iff (L \le c \le U \Rightarrow f(c) \le 0)$*, where c is the critical point of f.*

[5] We will not consider here the sign definite conditions but they can be treated in a similar fashion.

Lemmas 3.2 - 3.5 are useful if one can determine algorithmically when a function is monotone increasing/decreasing or convex/concave on a certain interval $[L, U]$. Checking algorithmically these for arbitrary degree f is a challenging problem since we have to compute the real roots of f' (or to find isolating intervals for them), which are the critical points of f. However, for degree 1 or 2 these checks can be performed easily (Lemmas 3.6 - 3.9).

Lemma 3.6. *Let f be a polynomial function of degree 1 on y. If the leading coefficient of f on $[L, U]$ is positive then f is increasing on $[L, U]$.*

Lemma 3.7. *Let f be a polynomial function of degree 1 on y. If the leading coefficient of f on $[L, U]$ is negative then f is degreasing on $[L, U]$.*

Lemma 3.8. *Let f be a polynomial function of degree 2 on y and c its critical point.*

(a) If $c < L \leq U$ and the leading coefficient of f on $[L, U]$ is positive then f is monotone increasing on $[L, U]$.

(b) If $L \leq c \leq U$ and the leading coefficient of f on $[L, U]$ is positive then f is convex on $[L, U]$.

(c) If $L \leq U < c$ and the leading coefficient of f on $[L, U]$ is positive then f is monotone decreasing on $[L, U]$.

Lemma 3.9. *Let f be a polynomial function of degree 2 on y and c its critical point.*

(a) If $c < L \leq U$ and the leading coefficient of f on $[L, U]$ is negative then f is monotone decreasing on $[L, U]$.

(b) If $L \leq c \leq U$ and the leading coefficient of f on $[L, U]$ is negative then f is concave on $[L, U]$.

(c) If $L \leq U < c$ and the leading coefficient of f on $[L, U]$ is negative then f is monotone increasing on $[L, U]$.

We implemented the results presented above in Mathematica (Simplifier routine `http://www.risc.jku.at/projects/SPy/CASC2016/`). It *(a)* uses the tactics for eliminating a quantifier from Lemmas 3.2 - 3.5 if their preconditions are fulfilled (Lemmas 3.6 - 3.9) *(b)* applies `Reduce` command of Mathematica to eliminate the rest of the quantifiers. It was successfully applied to all, but one, quantifier elimination problems appearing in the synthesis of optimal algorithms (Section 3.1.2), however *(a)* in some cases the quantifier-free formula obtained by Mathematica was further simplified by hand for esthetic reasons *(b)* in the case the quantifier-free formula could not be found by Mathematica, we manually eliminated one variable then applied QEPCAD-B, concluding that the formula simplification step of cylindrical algebraic decomposition plays a major role at delivering the final answer.

Application: Synthesis of Optimal Numerical Algorithms

In this section we only state the main result. The proof, similar to [48, 48], can be checked in [45].

Theorem 3.10 (Main). *We have*
(A) $C \wedge Q \wedge 0 < E < 1 \quad \Longrightarrow \quad E \geq \frac{1}{4}$
(B) $C \wedge Q \wedge E = \frac{1}{4} \quad \Longleftrightarrow \quad p = (-1, 0, 0, 1, 1) \wedge q = \left(-\frac{3}{4}, -\frac{1}{2}, \frac{1}{4}, 1, 1\right)$

3.1.3 Conclusions

In this section, we presented combined real quantifier elimination techniques and simplification techniques for the synthesis of optimal numerical algorithms. The problem formulation is based on static analysis formal methods approach: given the general template of numerical algorithm, we ensure that it is correct and it terminates. This is expressed as quantified first order logical formulae under the real domain and solved using symbolic computation techniques, namely quantifier elimination by cylindrical algebraic decomposition. However, state-of-the-art general quantifier elimination techniques are not able to solve such complex problem, so we manually devised simplification techniques based on the underlying anatomy of the quantified formulae. We also automated some of these methods.

Future research directions could be:

- fully integrate the simplification techniques into state of the art quantifier elimination tools.
- synthesize optimal numerical algorithms which compute the n^{th} square root of a real number.

3.2 Formal Methods in Cloud Computing

This line of research is the most extensive I have worked on after the PhD studies. It started while working on the SPECS project and continued in the projects MANeUveR and SAGE.

The most important contributions in this area are as follows.

1. The prevalence of Cloud services introduces security risks, particularly when unclear security policies and misalignment with user needs exist. Security Service Level Agreements (Security SLAs) are essential for secure Cloud adoption, but their practical use is hindered by enforcement challenges. In [32], we presented a method to establish a service catalog of security capabilities for as-a-service delivery, aiding providers in offering and customers in assessing monitored security features through Security SLAs. We briefly introduce the paper in Section 3.2.1.

2. Addressing the complex challenges of provisioning Cloud services under Security SLAs, in [31], we focused on representing understandable and measurable security features, automating security mechanism provisioning, and continuously monitoring services. We introduced a framework that enhances cloud applications with security features, presenting a novel Security SLA model

and a security-driven planning process. This process considers security component implementation constraints, customer requirements, and enables automatic resource provisioning, illustrated through a practical real-world case study. A brief presentation of the paper is in Section 3.2.2.

3. By generalizing the Planning component introduced in [31] which has the role of resource provisioning, we automated the deployment of component-based applications in the Cloud which involves allocating virtual machine (VM) offers to meet component interactions, hardware/software requirements, and performance goals. Typically a complex optimization challenge, in [51, 38, 52], we addressed the issue by systematically analyzing the problem's specifics and devising diverse exact strategies (constraint programming, mathematical programming, SMT solving) for reducing the search space. By leveraging symmetries, graph representations, and their synergies, the study achieves a scalable deployment solution, particularly employing a variable reduction strategy alongside a column-wise symmetry breaker to effectively optimize the intricate VM offer selection problem. In [82], we approached the same problem by using approximated methods, a population-based metaheuritics. The results are briefly presented in Section 3.2.3.

3.2.1 Automatically Enforcing Security SLAs in the Cloud

The rapid and widespread integration of Cloud computing across various domains has necessitated the introduction of Service Level Agreements (SLAs) to meet customer demands, especially for security-related SLAs (Security SLAs). The focus on Security SLAs is relatively recent, leading to ongoing research efforts to advance the understanding and implementation of such agreements (see [41] for a survey). Contemporary security enforcement strategies, both in traditional and Cloud environments, often rely on certification approaches (e.g. NIST [64], Cloud Security Alliance[6]), assuming comprehensive knowledge of service layers and employing static security configurations verified through audits. In contrast, Security SLAs demand the automated setup and configuration of security features based on customer requirements.

Addressing this challenge requires a clear representation of security needs understandable to both customers and providers, alongside effective monitoring of verifiable security-related Service Level Objectives (SLOs). This complex landscape, identified as a research challenge (see, e.g., papers [41, 18], as well as the research papers SPECS[6], CUMULUS[7] and MUSA[8]), prompts the exploration of an innovative Security SLA model. This model leverages the widely recognized WS-Agreement standard for SLA representation and management [8], enriched with security attributes and aligned with recent standardization initiatives from NIST and ISO. Notably, this model introduces automatic security policy enforcement by mapping user security requirements to a Security SLA that offers guaranteed security controls over time, even in the face of potential security incidents.

A critical innovation of our paper [31] is the incorporation of security capabilities

[6] https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v3/

[7] https://cordis.europa.eu/project/id/318580

[8] http://cordis.europa.eu/project/rcn/194208_en.html

and metrics to support measurable Security SLOs. These are enforced and monitored through the activation of appropriate security mechanisms and monitoring systems, guided by a standardized Security Control Framework embedded in the SLA model. The paper presents a novel solution that autonomously acquires and configures Cloud resources for the optimal deployment of security-related software components, aligning with Security SLOs.

Our approach hinges on matching customer security requirements in a Security SLA with available security mechanisms offered as a service (Security-as-a-Service), subsequently generating an allocation plan for deploying software components aligned with desired security mechanisms. This security-driven planning problem is intricate, encompassing deployment-specific constraints of security mechanisms and security-related constraints stipulated by Security SLOs. While resource allocation based on Quality of Service parameters is documented, comprehensive solutions for automated security provisioning based on SLAs are notably absent. Existing provisioning solutions focused on cost optimization and infrastructure utilization in High Performance Computing contexts cannot be directly transferred to the dynamic Cloud environment.

To demonstrate the viability of the proposed approach, a practical case study involving a Secure Web Container application provisioning is presented. The planning problem's application is showcased, detailing the entire enforcement phase leading to an optimal plan. The paper concludes by addressing potential planning anomalies and strategies to prevent them.

3.2.2 A Security SLA-driven Methodology to Set-up Security Capabilities on Top of Cloud Services

Cloud services have become a pivotal segment of the contemporary IT landscape. The widespread embrace of Cloud Computing has led numerous organizations and individuals to rely on these services for their operations. However, the utilization of third-party services, often shared among diverse customers, raises concerns about reduced control over personal data and sensitive information. This drawback negatively impacts Cloud Service Providers' (CSPs) business objectives and hampers the adoption of Cloud services, particularly for security-conscious customers. The limited transparency from providers regarding security and the resultant diminished customer confidence in the services stem from divergent perspectives on security between CSPs and customers. CSPs often express security guarantees using technical, intricate language that proves challenging for non-technical users to comprehend.

Recent research efforts, spanning academia, industry, and government-driven initiatives [33], have focused on Security Service Level Agreements (Security SLAs) and their application within Cloud environments. For identifying security parameters, guidelines and international standardization initiatives exist to establish a shared collection of security controls encompassing both technical and non-technical security aspects (e.g., ISO27002[9], NIST Security Control Framework [64], and Cloud Control Matrix from Cloud Security Alliance[6]). These frameworks serve to help organizations evaluate their services' security by specifying the enforced set of security controls (security capabilities). However, despite the keen interest in security and

[9] https://www.iso.org/standard/54533.html

standardization endeavors, CSPs predominantly emphasize performance-related parameters in their SLAs. As it stands, customers can merely accept services as offered without the means to negotiate, let alone monitor, the security level of the acquired services.

Paper [32] is partly grounded in the efforts of the EU projects SPECS and MUSA, focusing on developing SLA-based Cloud security services and fostering security-by-design in multi-cloud contexts through Security SLAs. Particularly, it introduces a practical methodology that aligns customer-defined requirements with provider-offered capabilities, leveraging existing security control frameworks' guidelines. This methodology serves as a guide for providers aiming to offer security features alongside guarantees. The novelty of this contribution lies in explicitly considering the constraints imposed by formal security guarantees, necessitating the identification of appropriate metrics and corresponding Service Level Objectives (SLOs) to enforce and monitor requirements during system operation.

3.2.3 Benchmarking Optimization Solvers and Symmetry Breakers for the Automated Deployment of Component-based Applications in the Cloud

In this section we present the summary of the results [51, 52, 82]. Papers [51, 82] are the preliminary ones, the most mature one is [52] which is a journal paper. In fact, in [38], we extended the work from [52] but we proved that the results from [52] regarding the best combination of tool, i.e. Z3 [83], and symmetry breaker, i.e. FVPR, are not improved.

The problem of *automated deployment* in the Cloud of component-based applications received attention due to increased demand of digitalization of businesses. It consists of the following steps: *(1)* selection of the computing resources, *(2)* the distribution/assignment of the application components over the available computing resources, and *(3)* its dynamic modification to cope with peaks of user requests. In paper [52], we tackled the first two steps of the deployment problem. In particular, our approach was used to synthesize the initial static optimal deployment of the application which consists of an assignment of application components to VMs such that the application functional requirements are fulfilled and costs are minimized.

The contributions of [52] are:

1. we formalized the Cloud deployment problem by abstracting the particularities of four classes of real-world problems;

2. we proposed a methodology analyzing the particularities of the problem with the aim of identifying search space reduction methods (these are methods exploiting the symmetries of the general Cloud deployment problem, respectively methods utilizing the graph representation of the interaction between the components of each application);

3. we assessed and compared the performance of two types of tools, namely mathematical programming (CPLEX [105]) and computational logic (the optimization modulo theory solver Z3 [39]);

4. we identified limits in their scalability and applied six search space reduction methods aiming to improve their performance.

Contributions of the work presented in this section, extension of [52], also presented in the SC-square 2022 workshop[10], are:

1. a new formalization in the Minizinc [88] constraint modeling language;

2. the performance comparison adds the constraint programming solvers OR-Tools [89], Gecode [56] and Chuffed [35] which are available from the Minizinc IDE;

3. the list of the symmetry breakers from [52] is enriched with composition of all possible combinations of single symmetry breakers. These symmetry breakers are tested on the constraint programming solvers OR-Tools [89], Gecode [56] and Chuffed [35], optimization modulo theory solver Z3 [25], and mathematical programming solver CPLEX [105].

Setting the Scene

Problem Definition The description of the problem first appeared in [52]. We have N interacting components, $C = \{C_1, \dots, C_N\}$, to be assigned to a set of M virtual machines, $V = \{V_1, \dots, V_M\}$. Each component C_i is characterized by a set of requirements concerning the hardware resources. Each virtual machine, V_k, is characterized by a *type*, which is comprised by hardware/software characteristics and leasing price. There are also *structural constraints* describing the interactions between components. The problem is to find:

1. an assignment matrix a with binary entries $a_{ik} \in \{0, 1\}$ for $i = \overline{1, N}$, $k = \overline{1, M}$, which are interpreted as follows: $a_{ik} = 1$ if C_i is assigned to V_k, and 0, otherwise; and

2. the type selection vector t with integer entries t_k for $k = \overline{1, M}$, representing the type (from a predefined set) of each VM leased;

such that: *(i)* the structural constraints, and *(ii)* the hardware requirements (capacity constraints) of all components are satisfied; and *(iii)* the purchasing/ leasing price is minimized.

The *structural constraints* are *application-specific* and derived in accordance with the analysis of the case studies. These are:

- *Conflict:* components in conflict cannot be deployed on the same VM.

- *Co-location:* components in co-location must be deployed on the same VM.

- *Exclusive deployment:* Only one of the components in exclusive deployment must be deployed in the same deployment plan.

- *Require-Provide:* one component requires or provides some functionalities offered, respectively provides, of another. Such an interaction induces constraints on the number of instances corresponding to the interacting components as follows.

[10] http://www.sc-square.org/CSA/workshop7.html

- *Full deployment:* components in this relationship must be deployed on all VMs leased, except those which would induce conflicts between components.
- *Deployment with bounded number of instances* occur when the number of instances of deployed components must be equal, greater or less than some values.

General constraints are always considered in the formalization and are related to the: *(i) basic allocation* rules, *(ii) occupancy* criteria, *(iii)* hardware *capacity* of the VM offers, and *(iv) link* between the VM offers and the components hardware/software requirements.

We stated the problems as a linear constraint optimization problem (COP). We redirect the reader to [52] for a full description of it.

Problem Formalization The formalization for all three types of solvers, that is constraint programming solvers, SMT solvers, and mathematical programming solver, has almost a one-to-one correspondence between linear constraints present in the definition of COP and the implementation. We did not apply optimizations exploiting the particularities of the modelling languages because we wanted to have a fair comparison of the different formalisms. However, in the future work we plan to take advantage of their sweet spots.

Minizinc models The Minizinc models are the ones newly introduced in this paper. They are organized as follows: *(i)* there are surrogate models for each problem in which the maximum number of needed VMs is computed; *(ii)* there is a model gathering together all constraints, both general and application specific (the model corresponding to each application instantiates the constraints needed for its modeling) *(iii)* there is a model gathering together all symmetry breakers developed which are then instantiated based on the tests which want to be performed.

Experimental Analysis

The principles of the experimental analysis were introduced in [52]: on one hand, we want to assess the *scalability* of state-of-the-art general CP (Chuffed [35], Gecode [56], OR-Tools [89]), MP (CPLEX [105]) and OMT (Z3 [25]) tools in solving COPs corresponding to realistic case studies. On the other hand, we evaluate the *effectiveness* of various static symmetry breaking techniques in improving the computational time of solving these problems (see Section 3.2.3). This is because tests (see Tables 3.1-3.2) revealed that the naive application of general CP, MP and OMT techniques is not sufficient to solve realistic Cloud deployment applications.

We consider four case studies (Secure Web Container, Secure Billing Email Service, Oryx2, and Wordpress) which exhibit: *(i)* different hardware characteristics of components and the rich interactions type in between (structural constraints); *(ii)* the kind of linear constraints used to formalize the problem; and *(iii)* the kind of solution we are searching for. A full description of these case studies is in [52].

The scalability and effectiveness are evaluated from two perspectives: number of VM offers, respectively number of deployed instances of components. For *Secure Web Container*, *Secure Billing Email* and *Oryx2* applications, we considered up to 500 VM offers. Additionally, for the *Wordpress* application, we considered up to 12 instances of the Wordpress component to be deployed. The set of offers was crawled from the Amazon CPs offers list.

Experimental Settings

Selected Symmetry Breaking Strategies Aiming to reduce the search space size, a set of strategies have been selected in order to exploit the particularities of the problem: *(i)* the VMs needed for application deployment might have different characteristics; *(ii)* applications components might be in conflict hence conflict-type constraints can be exploited; *(iii)* the number of instances to be deployed is unknown.

Our approach is incremental and experimental: we start with traditional symmetry breakers that have been used for other problems related to bin-packing and combine them with the aim of further search space reduction.

Simple symmetry breakers

Price-based ordering (PR). This strategy aims to break symmetry by ordering the vector containing the types of used VMs decreasingly by price, i.e. $p_k \geq p_{k+1},\ k = \overline{1, M-1}$. This means that the solution will be characterized by the fact that the columns of the assignment matrix will be ordered decreasingly by the price of the corresponding VMs.

Lexicographic ordering (LX). This corresponds to the traditional strategy aiming to break column-wise symmetries. The constraints to be added aiming to ensure that two columns, k and $(k+1)$ are in a decreasing lexicographic order, i.e. $a_{*k} \succ_{lex} a_{*(k+1)}$, are $\bigwedge\limits_{l=1}^{i-1}(a_{lk} = a_{l(k+1)}) \implies a_{ik} \geq a_{i(k+1)},\ \forall i = \overline{1, N}$.

Load-based ordering (L). This is a column-wise symmetry breaker which orders decreasingly the columns by the number of the component instances they accommodate: $\sum_{i=1}^{N} a_{ik} \geq \sum_{i=1}^{N} a_{i(k+1)},\ k = \overline{1, M-1}$.

Fixed values (FV). The search space can be reduced also by fixing the values of some variables starting from the application specific constraints, in particular conflict constrains. More precisely, the graph composed by the components being in conflict is used to identify components which must be placed on different machines and hence the values of the corresponding decision variables are fixed. The identification of these components is done by constructing the clique with maximum deployment size.

Composed symmetry breakers The symmetry breakers above can be composed leading to the following symmetry breakers:

- FV-PR, FV-L, FV-LX, PR-L, PR-LX, L-PR, L-LX,
- FV-PR-L, FV-PR-LX, FV-L-PR, FV-L-LX, PR-L-LX, L-PR-LX,
- FV-PR-L-LX, FV-L-PR-LX

These symmetry breakers are so the subsequent breaks ties of the former. For example, FV-PR fixes on separate machines the decision variables corresponding to the component instances being in the clique with maximum deployment size and the machines left unoccupied are ordered decreasingly by price. In the case of PR-L-LX, the machines are ordered decreasingly by price, those with the same price are ordered decreasingly by the number of components they host and those with the same number of instances are ordered lexicographically.

It is worth noticing that the symmetry breakers involving FV must apply FV the very first. This is because FV is used as a preprocessing step which has a positive impact on the solvers as it introduces equalities.

Software and Hardware Settings We used Minizinc v0.7.0 as the constraint modeling language. We mention that the Minizinc models follow the formalization and no optimizations were performed because we wanted to be as close as possible to the OMT and CPLEX formalizations in order to have a fair computational comparison between the newly considered solvers and the existing results. The CP solvers used (Chuffed, Gecode, OR-Tools) are called from Minizinc IDE with the default values for parameters. The OMT formalization is done using the Z3 Python API and uses quantifier-free linear integer arithmetic. Z3 was used with the default values of the parameters. In the case of the mathematical programming solver CPLEX, we used the Python API with the no symmetry breaking option manually activated.

The source code and the experimental results are available online[11]. All reported timings are in seconds. They only include the actual solving time of the optimization problem and not the pre-processing steps.

All tests in this paper were performed on an Intel(R) Core (TM) i5-9400F CPU @ 3.90GHz using Chuffed v0.10.4, Gecode v6.3.0, OR-Tools v9.0.0, CPLEX v12.9.0 and Z3 v4.10.2.

Results

Tables 3.1-3.4 include the results obtained without using symmetry breaking strategies. The list of offers (columns #o) was crawled from the Amazon site[12]. Each list of VM offers covers the main instance types, for example, `small`, `medium`, `large`. The list of offers can be viewed as a containment hierarchy (i.e. the list of 20 offers is included in the list of 40 offers etc.).

The tables include only those cases for which we obtained a result in a 40 minutes timeframe. The missing values (_) mean that no solution is returned in this timeframe.

One can observe that CPLEX scales the worst for all case studies while OR-Tools the best. However, none of the tools scale for Wordpress with more than 6 instances and several dozens of offers.

To overcome the lack of scalability issue, we applied the symmetry breaking strategies described in Section 3.2.3.

[11] `https://github.com/BogdanD02/Cloud-Resource-Provisioning`, release v1.0.0

[12] `https://aws.amazon.com/`

Table 3.1: Scalability tests for Wordpress with different instances (#i). Time values are expressed in seconds.

#i	**#o=20**	**#o=40**	**#o=250**	**#o=500**	**#o=20**	**#o=40**	**#o=250**	**#o=500**
OR-Tools					**CPLEX**			
3	3.49	8.38	96.05	191.04	9.66	-	-	-
4	23.25	56.07	501.43	987.91	121.96	-	-	-
5	149.47	425.03	-	-	446.02	-	-	-
6	493.39	1173.46	-	-	664.68	-	-	-
Gecode					**Chuffed**			
3	2.13	2.19	-	-	2.05	3.7	45.88	447.56
4	14.84	23.83	-	-	23.73	114.18	1866.61	-
5	162.13	-	-	-	531.19	2278.76	-	-
Z3								
3	2.82	4.13	103.19	391.87				
4	46.46	275.81	-	-				

Table 3.2: Scalability tests for Oryx2. Time values are expressed in seconds.

#o=20	**#o=40**	**#o=250**	**#o=500**	**#o=20**	**#o=40**	**#o=250**	**#o=500**
OR-Tools				**CPLEX**			
0.95	1.12	4.25	5.79	0.16	0.54	-	-
Gecode				**Chuffed**			
70.99	104.54	234.27	465.72	128.67	154.11	294.25	396.58
Z3							
13.35	15.36	453.2	717.99				

Discussion of the Results

We conducted various tests involving 5 solvers and 15 symmetry breakers. The are publicly available[11].

We draw the following remarks:

1. Using the 15 symmetry breakers, out of the 5 solvers, the best one from the computational time point of view is Z3.

2. For the virtual best solver, that is Z3, the best symmetry breaker is FVPR. This is because we have many Wordpress files to be analyzed (40 files corresponding to Wordpress with 3 up to 12 instances and 20, 40, 250, 500 offers), compared to the other applications (4 files corresponding to 20, 40, 250, 500 offers for each of the other applications) for which reduction methods which exploit the graph representation, that is FV, is of benefit.

3. We also considered the best symmetry breaker for Z3 for each of the case studies. In case of Secure Web Container and Secure Billing Email Service applications the best is FVL, while for Oryx2 is FVLX. There is no surprise that symmetry breakers involving FV give best results, however for more reliable results we should consider more test cases for Secure Billing, Secure Web and Oryx2, since now there are only 4 files analyzed corresponding to different number of offers. We plan to run more tests for each of the case studies for a more accurate analysis.

4. One would expect that the best symmetry breaker is one composing a higher number of individual symmetry breakers as more symmetries are broken so the search space is significantly reduced. However, this is not true: FVPR,

Table 3.3: Scalability tests for Secure Billing Email Service. Time values are expressed in seconds.

#o=20	#o=40	#o=250	#o=500	#o=20	#o=40	#o=250	#o=500
OR-Tools				**CPLEX**			
0.85	1.22	8.04	18.68	0.73	4.73	378.47	-
Gecode				**Chuffed**			
0.73	1.42	76.63	119.32	0.79	0.82	1.35	3.11
Z3							
0.34	0.70	3.37	8.29				

Table 3.4: Scalability tests for Secure Web Container. Time values are expressed in seconds.

#o=20	#o=40	#o=250	#o=500	#o=20	#o=40	#o=250	#o=500
OR-Tools				**CPLEX**			
1.50	4.85	28.79	57.33	2.68	49.64	-	-
Gecode				**Chuffed**			
10.15	43.84	76.63	779.44	0.89	1.01	2.64	4.54
Z3							
0.45	1.14	8.28	17.48				

composing 2 symmetry breakers, is better than those composing 3 or 4. An explanation for this is, on one hand the number of added constraints which influence the solving time, on the other hand, when using static symmetry breaking, the symmetry breakers can interact badly with the SMT solvers which we used as black box.

3.2.4 Conclusions

We presented our pivotal contributions to the realm of Cloud service security and provisioning. Firstly, it introduces a method to establish a comprehensive service catalog of security capabilities, aiding secure Cloud adoption through clarified security policies. Secondly, a framework enhances cloud applications by automating security provisioning and introducing a novel Security SLA model, effectively aligning security features with user needs. Lastly, the paper addresses the complexity of deploying component-based applications in the Cloud, presenting diverse strategies for optimizing virtual machine allocation.

Regarding future work, we will focus on the extensions of the results regarding Cloud provisioning. As detailed in Section 4.2, we plan:

- to further extend the symmetry breaking strategies and
- come up with template learning methods based on graph neural networks.

3.3 Formal Methods for Data-intensive Applications

The results presented in this section are the outcome of the collaboration within the project DICE (Developing Data-Intensive Cloud Applications with Iterative Quality Enhancements).

Big Data is an important field that explores innovative solutions to support data-intensive applications (DIAs), involving powerful software and hardware infrastructures to process vast amounts of information. The area has gained significance due to widespread applications like Facebook and Twitter. Developing methodologies and frameworks to leverage Big Data technologies for DIAs is now essential.

Our long term goal is the representation, at design-time, of the runtime behavior of Apache Storm[10] topologies. Apache Storm is a free and open source distributed realtime computation system which is used in DIAs, e.g. realtime analytics, online machine learning, continuous computation, distributed RPC, ETL. The notions used in Storm are as follows. A *topology* provides an abstract representation of a DIA through directed graphs, where nodes are of two kinds: *computational nodes* (named bolts in Storm) implement the logic of the application by elaborating information and producing an outcome, whereas *input nodes* (named spouts in Storm) bring information into the application from the environment. The aim is avoiding as much as possible the cases when applications need a re-design after their deployment. One way to achieve this is to analyze the safety of DIAs, which in terms of Storm technology is to analyze the safety of the topologies. Safety verification aims to verify that undesired configurations will not occur in the system.

We approach the safety verification using two formalisms.

- The first [78] is based on CLTLoc temporal logic [22]; the verification is limited to applications with a *bounded number of threads* running in bolts. In the CLTLoc model the length of the queue associated with a bolt might become infinite if the bolt cannot timely process the incoming stream. The verification properties which can be formulated and verified by this formalism refer to checking if the variables representing the queue size grow unboundedly.

- the second [20] is based on array-based systems approach [59] implemented in state-of-the-art tools MCMT[13] and Cubicle[14] for the modeling of Storm applications. This approach is more general as it allows the verification of *coverability properties*, e.g. checking if, given queue(s) bound(s) defined by the designer, “all bolt queues have a limited occupation level”.

In this section we briefly present three contributions related to the verification of DIAs. We direct the reader to the corresponding paper for detailed infomration:

1. The first [78] outlines an automated formal verification approach for distributed systems using Storm technology. This approach involves creating a formal model of Storm topologies using CLTLoc metric temporal logic extended with counters. A tool assists in generating formal models from high-level descriptions of the topologies, and the Zot formal verification tool checks desired properties related to queue growth in the system. Experimentation with example topologies demonstrates the impact of timing features on node queues. This result is presented in Section 3.3.1.

2. The second [20] focuses on formalizing and automating the verification of data-intensive applications built on Storm technology, a pioneering streaming framework. We utilize the array-based systems formalism introduced in [60],

[13] http://users.mat.unimi.it/users/ghilardi/mcmt/

[14] http://cubicle.lri.fr/

a suitable abstraction for modeling the runtime behavior. This formalization involves quantified first-order logic to symbolically represent array-based systems and verify safety properties using the state-of-the-art Cubicle model checker. This approach is presented in Section 3.3.2 along with challenges and limitations we encountered during work.

3. The third [19] highlights the increasing popularity of quality-driven frameworks for data-intensive applications, in line with the widespread adoption of Big Data approaches. The DICE framework aims to provide innovative tools and approaches for data-aware quality-driven development. Among these tools is the DICE Verification Tool (D-VerT), enabling designers to assess their designs with regard to safety properties, such as avoiding undesired system configurations. We present the D-VerT, available as an open-source resource on GitHub[15] in Section 3.3.3. D-VerT implements the verification of Storm topologies using both CLTLoc metric temporal logic and array-based systems formalisms.

3.3.1 Towards the Formal Verification of Data-Intensive Applications Through Metric Temporal Logic

Big Data is a prominent area, involving both academia and industry, researching innovative solutions to support the entire life-cycle (from design to deployment) of so-called data-intensive applications (DIAs), which are able to process huge amounts of information. The area gained considerable relevance in the recent years, especially promoted by the pervasive spread of social applications like, for instance, Facebook or Twitter. The term “Big Data” was coined almost ten years ago with the intent of referring to all those applications and frameworks requiring high computational power, commonly offered in private or public clouds, the latter providing their infrastructure as a service (IaaS). In this sense, Big Data applications are classified as software and infrastructures which manipulate a very diversified set of data in a relative small amount of time. Hence, defining frameworks for the development of DIAs that leverage Big Data technologies is nowadays of major importance.

The DICE project [30] defines techniques and tools for the data-aware quality-driven development of DIAs that leverage Big Data technologies hosted in clouds. It offers tools for quality assessment, architecture enhancement, agile delivery and continuous testing of DIAs. DIA designers are supported in developing high-quality applications that can be continuously deployed to satisfy service-level agreement (SLA) requirements formulated by final customers in terms of efficiency (e.g. of performance and cost-effectiveness), reliability and safety. In the DICE approach, designers model DIAs through UML diagrams tagged with suitable annotations capturing the features of Big Data applications, and in particular their *topology* by conducting a three-step refinement process. First, the application is structured by identifying the role of its components, without specifying the technological solutions that will be employed to realize their functionality. The obtained UML diagram is refined by defining the technologies for implementing each component. Finally, the model is further refined to specify the deployment parameters constraining the underlying cloud framework where the application actually runs.

[15] `http://dice-project.github.io/DICE-Verification`

A topology provides an abstract representation of a DIA through directed graphs, where nodes are of two kinds: *computational nodes* implement the logic of the application by elaborating information and producing an outcome, whereas *input nodes* bring information into the application from the environment.

The semantics underlying the topology typically changes depending on the target Big Data technology. In this paper we focus on the Apache Storm[10] technology—in which computational nodes are called *bolts*, and input nodes are called *spouts*—a framework which is widely used in applications that need reliable processing of unbounded streams of data, e.g. Groupon[16], The Weather Channel[17], Spotify[18]. In Apache Storm applications, one of the key concerns is that time-related parameters such as emission rates of data do not induce an excessive load on the topology by accumulating data in nodes' queues. The latest version of the framework offers options to adapt these parameters at run-time (e.g., by slowing down the input nodes) to mitigate the issue, but this might negatively and unpredictably impact other features of the application. Hence, one would like to design the topology from the beginning in a way that run-time adaptation is not necessary. An important feature of Apache Storm, which is available since version 1.0.0, is the automatic backpressure mechanism which mitigates the effects of an excessive load on the topology by slowing down the spout emission rate. This prevents the saturation of the queues of messages associated to bolts but does not guarantee that all the timing requirements are met, since the excessive load will be simply moved to the data source of the topology. Therefore, it is of huge importance that at design time incorrect timing constraints are analyzed since they might lead to latency in processing information and unbounded memory growth. Streams are infinite sequences of *tuples*, i.e., atomic messages exchanged by nodes through a message system. Verification does not consider the cause of a node failure but its effect on the requirements of the application. After a node failure, the total delay that the topology requires to process one tuple (or a set of tuples) can exceed the maximum tolerated delay or the quality of the information processing may degrade (which is not considered in the paper). In some reliable stream processing systems, lost tuples are first re-emitted and then re-processed newly by the whole topology, with an inherent undesired delay, while this does not occur in unreliable topologies where failed tuples are simply disregarded. Underestimating the computational power of nodes also affects the time to process tuples, which then might not meet the timing requirements, as the processing delay might cause the saturation of queues of the message system. The analysis does not take into account the quality aspect of the processing, and it focuses instead on the temporal aspects of the implemented topology.

In this paper, we approach such design with three contributions.

1. We define a formal model of DIAs based on the Storm technology. This model, which we call the *timed counter networks* model, is expressed through the Constraint LTL over clocks (CLTLoc) [23] metric temporal logic enriched with positive counters. CLTLoc allows users to express time delays, and the addition of positive counters allows for the description of memory usage issues such the evolution of the length of nodes' queues.

[16] https://www.groupon.com/

[17] https://www.weather.com/

[18] https://www.spotify.com/

2. We allow for the automated verification of such formal models through the D-VerT (DICE Verification Tool) prototype tool, which is based on the Zot bounded satisfiability checker [4]. By performing formal verification tasks through D-VerT, designers can detect bad configurations producing undesired consequences, such as data processing delays causing an unbounded use of memory. Therefore, our technique takes advantage of bounded verification approaches, which allow for the identification of bad executions through the exploration of finite prefixes of topology executions representing infinite runs violating a property.

3. We define sufficient conditions for guaranteeing the soundness of the verification results obtained through D-VerT. In fact, the extension of CLTLoc with unbounded counters makes the logic undecidable in general, so we must guarantee that the conditions and abstractions introduced to make the verification technique applicable in practice do not generate spurious results.

3.3.2 Formal verification of data-intensive applications through model checking modulo theories

The research presented in the paper [20] and outlined in this section has the same purpose as the one in [78], namely the *safety verification of DIAs, i.e. undesired configurations will not occur in the system* modeled in Apache Storm technology, however it is more general. More precisely, in [78] we expressed the verification problem with a different formalism, namely the CLTLoc temporal logic [21]; the verification was limited to applications with a *bounded number of threads* running in bolts. The model introduced in [20] has the ability to represent an *arbitrary number of processes* (parameterized infinite-state model checking). This feature can be used to analyze an arbitrary level of parallelism in DIA components such as Storm bolts.

Our contributions of using model checking modulo theories for the formal verification of DIAs are as follows.

The *first contribution* on this topic is the first application of array-based systems to model and verify parametric aspects of DIAs in an infinite-state paradigm. Array-based systems have been successfully applied for the formalization and verification of parameterized timed systems, fault tolerant systems and imperative programs, but never to DIAs.

The *second contribution* is the understanding that the lack of a proper abstraction to model arbitrary (yet finite) processes in Storm components impedes the identification of a meaningful verification problem that can be defined in terms of safety verification of an array-based system.

The *third contribution* is a set of lessons learned concerning the state-of-the-art tools, which do not support specific features for limiting the state space exploration with user-defined criteria that would promote a reduction of time and memory required for the analysis. In fact, the last model was obtained after many steps of refinement needed to overcome the limitations of the tools.

In [78], we verified the property that "none of the bolt queues can grow unboundedly". In this work, the verification problem is a *coverability analysis*, checking whether, given queue(s) bound(s) defined by the designer, "all bolt queues have a limited occupation level".

We do not focus on modeling and verifying the ordering of the tuples in the queue, but rather on how the quantities changes. This is because distributed systems have unreliable communication hence messages can be lost. In this respect, our formalism should be seen as lossy vector addition systems [27] augmented with timing constraints along system executions through the notion of clocks.

Preliminaries

In this section we provide a brief overview of the Apache Storm technology and of the formalism chosen to model applications based on this technology.

Apache Storm is a stream processing system that allows parallel, distributed, real-time processing of large-scale streaming data on scalable systems. The key concepts in Storm applications are *streams* and *topologies*. Streams, called tuples, are infinite sequences of string-based messages that are processed by the application. Topologies are directed graphs of computation, whose nodes correspond to the operations performed over the data flowing through the application, and whose edges indicate how such operations are combined, i.e., the streaming paths between nodes.

There are two kinds of nodes, *spouts* and *bolts*. Spouts are stream sources that get data from external systems such as queuing brokers or web APIs. Bolts apply transformations over the incoming data streams and generate new streams to be processed by the connected bolts. Connections are defined at design time by the subscription of the bolts to spouts or other bolts. The number of processes running in parallel for each node can be arbitrary (unbounded *parallelism*).

Array-based Systems [59] are the mechanism we exploit to formalize the behavior of Storm topologies, as their state can be seen as a set of unbounded arrays whose indexes range over the number of processes of the nodes (the number of nodes in a Storm application is fixed and never changes). Array-based systems are symbolically represented by a class of quantified first-order formulae whose satisfiability is decidable under reasonable hypotheses on the theories of indexes and elements. An array-based system is a tuple $S = (A, Init, \tau)$ where A is a set of function symbols (arrays) representing the state variables, $Init$ is a formula which characterizes the initial states of the system (in which the variables from A can appear free) and τ is a transition relation. Transition relation τ is expressed as a disjunction of existentially quantified formulae, where each disjunct is a parameterized transition of the system. A transition t relates the array variable a with an updated variable a' and has the form $\underset{\bar{i}}{\exists}\, G(\bar{i}, A) \wedge \bigwedge_{a \in A} \underset{\bar{j}}{\forall} a'(\bar{j}) = U_a(\bar{i}, \bar{j}, A)$. G is called the guard of t, U_a the update of a, and the conjunction of the formulae following G is the action of the transition t. Safety properties are expressed by characterizing unsafe states. An unsafe formula is a conjunction of literals l_k, $k = \overline{1, n}$, existentially quantified: $\underset{\bar{i}}{\exists}\, l_1(\bar{i}) \wedge ... \wedge l_n(\bar{i})$. The formalization of an array-based system consists of the set of initial states, the ordering of the actions (by means of a transition relation) and the set of unsafe states.

Array-based systems can be formally verified through a decision procedure based on *backward reachability*, by repeatedly computing the pre-images of the set of unsafe states (obtained by complementing the property to be verified). The analysis halts in two cases, either when the current set of reachable states has a non-empty intersection with the set of initial states (*safety check*) – and the system is unsafe,

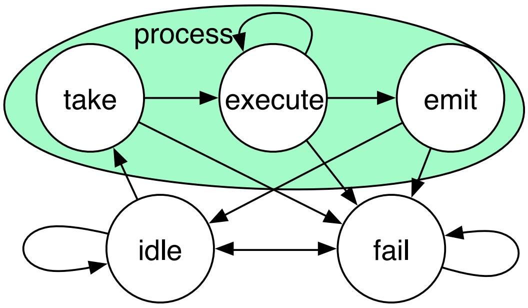


Figure 3.3: Finite automata describing bolts behavior.

or when such a set has reached a fix-point (*fix-point check*), i.e. further application of the transition does not enlarge the set of reachable states – and the system is safe.

Modelling Assumptions

This section describes the model of Storm topologies. It focuses on the behavior of the queues of the bolts of Storm topologies and describes how the timing parameters of the topology, such as the delays with which tuples are input to the topology by spouts and the processing time of tuples for each bolt, affect the accumulation of tuples in the queues. To this end, our models capture the *progress of time* in the system and make use of discrete counters to describe the evolution of the *size of the queues*.

Some of the assumptions for the formalization are the same as in [78]. We do not consider deployment details, such as, for instance, the number of worker processes and the underlying architecture. Spouts do not have any input queue or incoming connection from other bolts, whereas bolts have internal queues, of unbounded size, which store the incoming tuples received from subscribed nodes of the topology. The various modeling exercises in this work used two different way for abstracting a bolt: the one where each bolt has one receiving queue for each of its parallel instances and the other one where only one single receiving queue is shared among all its parallel instances. In any case, no sending queue is represented.

We do not detail the contents of tuples, but only their quantities, since the verification problem is on the size (number of tuples) of the queues. Spouts are considered sources of tuples and their queues are not represented.

A Storm topology is a directed graph where the set of nodes includes the sets of spouts and bolts. Our models represent a topology through by encoding the subscription relation among bolts and spouts. The behavior of bolts can be illustrated by means of finite state automaton (see Figure 3.3). The automaton is the same as in [78]. However, some limitations of the current model checking modulo theories approaches required us to simplify the automaton and remove state *take*.

Moreover, for simplicity of the modeling, we do not model bolt failures; then state *fail* will not be part of the model. On the other hand, spout failures are not modeled on purpose; their effect is irrelevant for the growth analysis of bolt queues as they would reduce the workload on the topology.

Timing parameters of a topology are: a) the time required by bolts to process a tuple (which is called *Execution rate*); b) the minimum time between two consecutive spout emits. We do not consider the maximum time because we can assume that a topology with sparse spout emits (with emit time tending towards infinity) is safe.

Formalization and Verification

The duration of the entire processing is kept in the real variable T representing the global clock. The status of the bolt with index i of process j is indicated by variable $B[i, j]$, which can be equal to: (`E`)*mit*, (`I`)*dle* and *E*(`X`)*ecute* (*state Ta*(`K`)*e* was initially considered but omitted in the end). The length of the queue associated to the bolt is indicated by an array variable L. The percentage of the processing of the tuple being elaborated by process j of bolt i, since the last take, is indicated by variable $P[i, j]$. The elapsed time from the last emit action performed by spout i is indicated by an array variable *Stime*. The constant Ts_{min} is a parameter for the system and represents the minimum time between two consecutive spout emit performed by the same bolt. The subscription relation among bolts and spouts and among bolts is indicated with the array variables *SubscribedBS* and *SubscribedBB* of booleans, respectively. Predicate $\mathit{SubscribedBS}[i, j]$ indicates that "bolt i subscribes to the streams emitted by spout j" while $\mathit{SubscribedBB}[i, j]$ indicates that "bolt i subscribes to the streams emitted by bolt j".

Verification problem. The safety property requires that all the bolts have bounded queue, i.e., the size of their queue does not overflow a given constant. Verification is achieved by checking if a state satisfying the negation of the property can be reached from the initial state. The formula encoding the property varies from model to model as it depends on the queue modeling. The formula for the *shared queue* case is $\underset{i}{\forall} L[i] < c$, with c a constant value.

Modeling topologies. We chose to model the evolution of a topology with both discrete and continuous transitions [7]. This allows us to differentiate the effect of the transitions, as they are fired in an alternating manner, and to simplify the model, as discrete transitions only enforce updates on state variables and not on the time T (the time elapsing affects the processing of tuples carried out by bolts). Discrete transitions have the purpose of either changing the state of the components or updating the size of the queues of the bolts but they do not modify the value of variable T. Conversely, the transition modeling the elapsing of time adds a positive amount δ to variable T. It possibly changes the states of some bolts when their processing has been terminated during the last δ time units. To model the alternation between the two types of transitions, we include in the model a flag called *CanTimeElapse* that, when positive, allows time elapsing transition to fire.

The graph of a topology invariants on the predicates *SubscribedBS* and *SubscribedBB*, as their value never change over the execution of the topology.

Time elapsing in the topology is expressed by (3.1). It exploits an array variable, called N_{proc}, which defines the number of active processes in bolts. The value of $N_{proc}[i]$, for any i representing a bolt in the topology, is not fixed *a priori* as it is an arbitrary parameter of the system. $P[i, x]$ stores the percentage of the tuple which still has to be processed. When process x receives a tuple and starts the execution, the value of $P[i, x]$ is set to 1 and keeps decreasing from 1 to 0 while time progresses due to (3.1). Formula (3.1) states that if there exists a positive value δ and $CanTimeElapse = \texttt{true}$ then: *(a)* the time progresses by incrementing T with δ time units; *(b)* for all the j and z such that δ is not big enough to complete the remaining part of the computation, i.e., $P[j,z] - \delta \geq 0$, then $P[j,z]$ is updated with the progress $P[j,z] - \delta$; otherwise, if δ allows the bolt to complete the current processing (when $P[j,z] - \delta < 0$) then $P[j,z]$ is set to 0. Observe that $P[j,z]$ is always ≥ 0 (by construction) and when it is non null then process z of bolt j is in execute state `X` and z is an index of an active process of the bolt i, i.e., $0 \leq z < N_{proc}[j]$.

$$\begin{aligned} &\underset{\delta}{\exists}\, 0 < \delta \wedge CanTimeElapse = \texttt{true} \wedge \\ &\underset{j,z}{\forall} \begin{pmatrix} T' & = T + \delta \\ P'[j,z] & = \textbf{if } (0 \leq P[j,z] - \delta)\textbf{then } P[j,z] - \delta \textbf{ else } 0 \\ B'[j,z] & \ldots \\ CanTimeElapse' & = \texttt{false} \end{pmatrix} \end{aligned} \tag{3.1}$$

Various refinements were undertaken to obtain the essential core set of transitions modeling the topology behavior. However, the rationale behind the following transitions is common to all versions of the model we have devised. We describe the three of them without details by outlining their functionality only. The set of transitions describing the behavior of the Storm topology allows state change according to Figure 3.3 and how this affects the accumulation of tuples in the queues: (1) $spout_{emit}(i,j)$: the queue of the bolt j subscribed to the spout i increases and the emit time of the spout is reset, (2) $bolt_{emit}(i,j)$: the state of bolt i is changed into idle and the length $L[j]$ of the queue of bolt j is incremented by 1, (3) $bolt_{take}(j,y)$: the length $L[j]$ of the queue of bolt j is decreased by 1 and the percentage of processing of the thread receiving the tuple $P[j,y]$ is set to 1. Note that we did not consider the state `K` but we actually model the take action through $bolt_{take}$.

3.3.3 D-VerT: a tool for verification of big-data applications

Big Data is an increasingly important field, devising solutions for DIAs using powerful infrastructures. Its relevance surged with the widespread use of applications like Facebook and Twitter. Defining methodologies and frameworks to develop such applications leveraging Big Data technologies is now fundamental. The DICE project addressed this challenge by introducing data-aware quality-driven development techniques, utilizing UML diagrams to model DIAs and their topology. The DICE Verification Tool (D-VerT) is a centerpiece of this approach, ensuring that the topology does not lead to undesired configurations, avoiding issues like processing latency and memory growth. By analyzing timing constraints, D-VerT safeguards against non-functional anomalies, ensuring robust and efficient Big Data applications.

Verification tool overview

Verification in D-VerT is conducted on annotated UML models, which encompass all crucial information about the topology. D-VerT receives a DICE Technology Specification Model (DTSM) in the form of a UML diagram with specialized stereotypes, capturing the technological aspects of the DIA platform chosen by the designer. The designer specifies the topology and relevant parameters, such as computational node

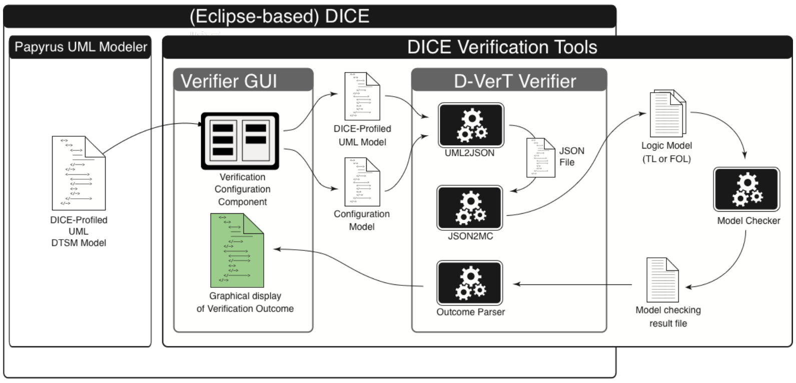


Figure 3.4: Verification workflow

processes or source node emit rate. A selected property is checked using templates in the DICE framework's IDE. The DTSM annotated model and property are transformed into a formal model for verification. D-VerT supports tailored verification approaches for various properties, generating and running the formal model with the chosen solver. The results are presented in the Verifier-GUI, indicating property validity and providing a trace if violated (see Figure 3.4).

Tool architecture

D-VerT consists of three modules (Figure 3.4):

1. DTSM2Json converts DTSM diagrams into a JSON file describing the topology,

2. Json2MC instantiates DIA semantics into a formal model in temporal logic or first-order logic, and

3. Outcome Parser displays the solver's results graphically on the IDE.

DTSM2Json extracts model features and user-provided configurations from the annotated UML file, serializing them into JSON. Json2MC is extensible and configurable, employing a Model Configurator and model templates to create formal models. The Outcome Parser visualizes verification results, such as counter-example traces illustrating system behavior (see Figure 3.5).

Verification approaches

D-VerT provides support for two verification approaches based on distinct logical formalisms. It offers an extensible architecture, allowing easy extension to other formalisms using parametric model-to-model transformations configured by templates.

The first approach, *bounded satisfiability checking*, involves two temporal logic formulae: one modeling the topology's temporal behavior and the other capturing the property for analysis. D-VerT checks if the conjunction of the former with the *negation* of the latter can be satisfied (i.e., if a violation exists). If not, the tool returns "unsatisfiable" (UNSAT); otherwise, it returns an execution trace as a counterexample. An UNSAT result implies the property holds in the model. This approach utilizes the Zot tool as the verification engine.

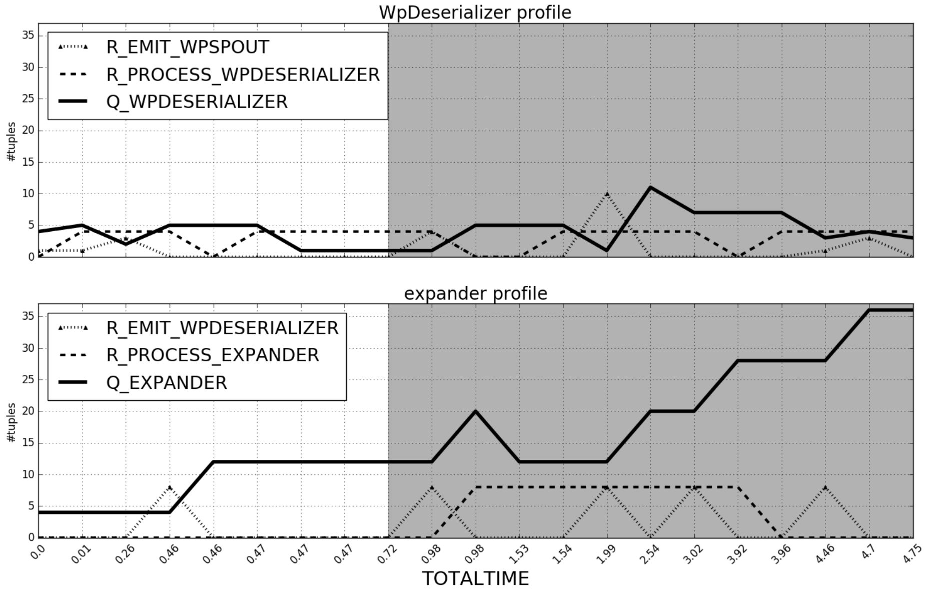


Figure 3.5: Example of D-VerT output trace

The second approach, *reachability checking*, defines a topology using an array-based system for safety verification. The model includes system transitions, an initial configuration, and a formula defining unsafe states. The result is either SAFE (no undesired configurations can be reached) or UNSAFE with an unsafe trace showing how undesired configurations can be reached. D-VerT employs MCMT[13] and Cubicle[14] for this verification approach.

Experimental Results

We utilized D-VerT, specifically the bounded satisfiability checking, to verify diverse topologies, including a complex one named ”focused-crawler” provided by an industrial partner in the DICE consortium. The property ”bounded occupation level of computational node queues” was validated to ensure timely information processing by bolts. A counterexample depicted an execution where at least one queue exhibited unbounded growth. Graphical traces from D-VerT illustrated periodic models with infinite repetitions after a finite prefix. By scrutinizing the traces, we confirmed the unbounded trend in the expander queue. Our experiments on various topologies and configurations evaluated the tool's performance, with results available on Github[15].

3.3.4 Conclusions

In this section we presented our achievements regarding safety verification of DIAs implemented in the Apache Storm technology. Firstly, an automated formal verification approach for distributed systems is outlined, employing Storm technology. This approach leverages CLTLoc metric temporal logic extended with counters to model Storm topologies, and a verification tool, Zot, to analyze queue growth properties. Secondly, formalization and automation of data-intensive applications on Storm technology are achieved, using quantified first-order logic and the Cubicle model checker. Finally, the work introduces the D-VerT tool within the DICE framework, enabling the assessment of system designs for safety properties, with the ability to verify Storm topologies through both CLTLoc metric temporal logic and array-based systems formalisms.

Planned future work is to extend the work using the model checking modulo theories approach [20]. At the time of working on that paper, the array-based systems formalism was not mature enough, hence, our formalization had to use different artifices to model the Storm topologies which made the verification cumbersome. Since then, the theory advance and we resumed our work on this topic.

3.4 Machine Learning

The contributions in this area refer to fake news detection and, respectively, training binarized neural networks for traffic signs classification. More precisely;

1. in [10] we propose supervised machine learning techniques for fake news detection, utilizing a dataset of fake and real news to train a model with `Scikit-learn` in `Python`. Employing text representation models like *Bag-of-Words*, *TF-IDF*, and *Bi-gram frequency*, the study explores *probabilistic* and *linear classification* for categorizing titles as *clickbait/nonclickbait* and content as *fake/real*. Results indicate that the linear approach combined with TF-IDF achieved optimal content classification, while Bi-gram frequency yielded lower accuracy for title classification compared to Bag-of-Words and TF-IDF.

2. in [91] we propose a bottom-up approach to design binarized neural networks (BNNs) architectures by analyzing constituent layers. Combinations of binarized convolutional layers, max pooling, batch normalization, and fully connected layers are explored with different parameters. Training on the German Traffic Sign Recognition Benchmark (GTSRB), proposed BNN architectures achieve over 90% accuracy on GTSRB and average over 80% accuracy on Belgian and Chinese datasets. Models maintain a parameter count from 100k to under 2M.

3.4.1 A Tool for Fake News Detection

In the current digital milieu, combating the proliferation of fake news has become paramount. Recognizing this, the European Commission established an expert group in 2018, emphasizing the importance of technological innovation in countering online misinformation. Our work contributes significantly by leveraging machine learning techniques to develop tools for detecting and reporting fake news articles.

The study's key achievements encompass the creation of advanced methodologies and tools that excel in identifying fake news. Drawing from the `Scikit-learn` library, probabilistic (*Naive Bayes*) and linear (*Support Vector Machine*) classification methods are harnessed alongside text representation models such as *Bag-of-Words*, *TF-IDF*, and *Bi-gram frequency*.

The pinnacle of this endeavor is the development of a robust fake news detection tool, exhibiting exceptional experimental results with an accuracy exceeding 80% for both content and title classification. Notably, the linear classification model paired with the TF-IDF model achieves remarkable accuracy at 94% for content classification. The study further explores title classification, with linear and probabilistic models yielding identical accuracy scores of 95%. Interestingly, the Bi-gram frequency model lags behind its Bag-of-Words and TF-IDF counterparts for title classification.

Problem Statement

Lots of factors influence the process of verifying and analyzing news articles, which makes it difficult to achieve a 100% accuracy. So we organised the factors we will be analyzing into: 1. source and the author, 2. title, 3. publication date, 4. content.

These factors combined together form the article full content. We observed that it was not enough to detect the content solely and ignoring the title, author and date. Every factor has its role in determining the credibility of a news article. For example, a news article can have a real content, but an exaggerated title to attract users to click on it. This requires title clickbait detection. Also, reposting old news as new one is common, so the date of publication is important too.

The analysis can be performed with computer science tools. At this aim, we implemented an algorithm (Algorithm 2) that takes as an input a link to a news article and, as output, it displays *details* about the entered article. These details are:

1. for *source and the author*: the news source type will be verified if it is credible or not, then a classification tag will be printed to the user; the author will be extracted and showed to the user; in case the author is not mentioned, the domain name of the website is considered the author.

2. for *title*: if the title is clickbait or nonclickbait.

3. for *publication date*: the most similar real news title that happened in the respective date.

4. for *content*: if the content is real or fake.

Our Approach

In this section we present algorithms for fake news detection. They are based on parsing and machine learning techniques. The main algorithm is Algorithm 2 which for every input link to an article displays information about the content, title, date and author by calling other subalgorithms. All these subalgorithms are self-explanatory. We will briefly describe the main ideas behind.

Parsing Web page parsing is the process of extracting information from a web page. Web parsing is based on the HTML source code of the web page. Through parsing, we were able to extract the required information for validation from the given web page, like title, content, date of publication, and author name (if exists).

Algorithm 2 Fake News Detector

```
input: Web link to a news article.
output: (1) Author: name/website. (2) Title: clickbait/nonclickbait. (3) Date:
the most similar news title in the respective publication date. (4) Content:
fake/real.
Step 1. Verify if the introduced link is trusted or not using http://www.opensources.co lists.
Step 2. If the introduced link is classified as trusted then go to Step 3. Else, print
a classification tag (fake, bias, etc.).
Step 3. Parse the HTML source code of the introduced link using Algorithm 3
and extract the following information from the web page: 1. Author of the article.
2. Publication date of the article. 3. Title of the article. 4. Content of the article.
Step 4.
    • Analyze the author using Algorithm 3; if the author name is missing then
      consider the website which published the article as an author.
    • Analyze the title, respectively content, using Algorithm 4; verify if the extracted
      title is clickbait/nonclickbait, respectively if the content is fake or
      real using machine learning and print clickbait/nonclickbait or fake/real accordingly.
    • Analyze the date using Algorithm 5; use the extracted publication date to
      check the news titles that actually happened in the respective date.
```

Machine Learning We used machine learning models to verify the article content and title. Using machine learning, we were able to: 1. check if the title is clickbait or not; 2. decide if the article content is fake or real.

In comparison with parsing, machine learning was used to solve classification problems. On the other hand, parsing was used for extracting data.

There are different types of machine learning algorithms, here we used supervised learning. Our algorithm takes a dataset as input; the output of the algorithm will depend on the input dataset. If the input is fake/real news dataset, then the output displays if the article is fake/real. On the other hand, if the input is clickbait/nonclickbait titles dataset, the output displays if the title is clickbait/nonclickbait.

Algorithm 3 Web Link Parsing

```
input: Web Link to a news article.
output: (1) The author of the article. (2) The publication date of the article.
(3) The title of the article. (4) The content of the article.
Step 1. Open the web link and get the HTML source code.
Step 2. Extract the title of the article.
Step 3. Extract the content of the article.
Step 4. Extract the publication date of the article.
Step 5. Extract the author of the article. If exists, then print the author. Else,
print the website domain name.
```

The machine learning process in our application consists of following steps:

1. In the case of content detection, we used the fake and real news dataset from `https://github.com/GeorgeMcIntire/fake_real_news_dataset`. In the case of title detection, we used the dataset mentioned in the paper [34]. These datasets are clean, labelled and ready to be used for features extraction.

2. We checked if words (tokens) in the articles and titles have a significant impact on whether the content was fake or real, and the title is a clickbait or not.

3. We chose the following text representation models: (a) Bag-of-Words model; (b) Term Frequency-Inverse Document Frequency model; (c) Term frequency Bi-gram model.

4. We implemented these three text representation models with two main classification approaches: linear and probabilistic.

5. We evaluated the classifiers using test data from the imported datasets. The test data were not used in the classifiers training process.

Cosine Similarity We used this approach to verify the similarity between the news title we have and a list of news titles. The list of news titles is extracted using the API `https://www.newsapi.org`. The list of titles is based on the date of publication we extracted from the news article web page. In other words, we have a list of titles for all events that happened at the same day the input article was published.

To perform the cosine similarity algorithm on the titles, we represented our given title and all the titles in the list as TF-IDF vectors. This means that every title we have has been represented as a vector of weights values for each word (token) in it. Now, we can perform the cosine similarity algorithm between our input title and every other title in the list. The highest similarity score will be recorded and its associated title will be showed to the user.

Algorithm 4 Analyze Article Content/Title

`input`: The article content/title
`output`: fake/real or clickbait/nonclickbait

Step 1. Read the dataset with fake and real news or with clickbait/nonclickbait titles and split it into train and test sets.

Step 2. Build the text representation model (Bag-of-Words, Term Frequency-Inverse Document Frequency, Bi-gram) from the train and test data.

Step 3. Fit the train data to machine learning classifiers: (1) `Naive Bayes` (Probabilistic classifier), (2) `Linear support vector machine` (Linear classifier).

Step 4. Predict the label (fake/real) of the article content or the label (clickbait/nonclickbait) of the article title using the machine learning classifiers.

Step 5. Use the test data from *Step 1* to calculate the accuracy score for the machine learning classifiers.

Algorithm 5 Analyze Publication Date

```
input: The article publication date.
output: News title with highest score similarity.
Step 1. If the date respects the ISO 8601 date format (yyyy-mm-dd or yyyy-mm-ddThh:mm:ss) then go to Step 3. Else go to Step 2.
Step 2. Transform the extracted date to the ISO 8601 date format.
Step 3. Get the list of news titles that happened in the respective date by sending a web request to https://www.newsapi.org.
Step 4. Build text representation model (Term Frequency-Inverse Document Frequency) from the extracted title and the list of received titles from newsapi.
Step 5. Use the Cosine Similarity approach to find the title which is the most similar to the extracted title.
Step 6. Print the extracted title, the other similar title and the similarity score.
```

Implementation Details

We implemented our application using Python (`https://www.python.org`) since it supports a large number of efficient packages helping to deal with any type of data (images, text, audio, etc.) and to achieve any target he wants (machine learning, deep learning, web development, etc.). To implement our application we used the following libraries: (1) `Scikit-learn` (`http://scikit-learn.org`), (2) `Pandas` (`https://pandas.pydata.org`), (3) `Beautiful Soup 4` (`https://www.crummy.com/software/BeautifulSoup/bs4/doc/`), (4) `PyQT5` (`https://www.riverbankcomputing.com/software/pyqt/intro`), (used for implementing the application's graphical user interface); and two external APIs: (1) Google Cloud Natural Language Processing API (`https://cloud.google.com/natural-language/`), (2) News API (`https://newsapi.org`).

Parsing In order to obtain the useful information from a given web link, our algorithm uses the Beautiful Soup 4 library for parsing the HTML source code and obtaining the title of the article and the content.

For extracting the publication date we did not use the Beautiful Soup 4 library, because there is no common HTML tag to represent the publication date. Hence we transferred the encoded HTML bytes obtained by the Beautiful Soup 4 object to string.Now, we can search for the date of publication by searching for any text that matches a date format pattern. We created a regular expression to extract all texts that match any possible date format.

For extracting the author name, since there was no specific and common HTML tag used to represent the author, we had to search for the author name in the HTML source code after we have converted it to string. To do that, we used the Google Cloud Natural Language Processing API to extract the entities of type *Person* from the whole HTML source.To get the author name, we had to search for the word *Author* in the source. The Google API returned all persons names from the given text, and this will be considered the author names. If the HTML source has no mention of the *Author* then the website domain name will be considered the author of the news article.

	BoW	TF-IDF	Bi-gram
MN	0.883	0.845	0.903
LSVC	0.883	0.941	0.868

Table 3.5: Content detection accuracy scores

Machine Learning

Analyzing Content and Title The verification of content and title is similar the only difference being the used dataset.

We started by reading the dataset using the `Pandas` library. Our datasets (for the content and title) are stored in two separated files of type csv. We have two data sets:

- fake_or_real_news.csv has a 6335 news articles, 3164 being fake, and 3171 being real. Every news article in the dataset is labelled as fake or real.
- log_32k.csv has 32000 titles, 15999 being clickbait, and 16001 being nonclickbait. Every title in the dataset is labelled as clickbait or nonclickbait.

After we have read the datasets, we used `Scikit-learn` library for all the steps involved in the supervised learning algorithm for fake news detection.

Analyzing Date of Publication For this analysis we used `https://newsapi.org` API and cosine similarity method from `Scikit-learn`.

Experimental Results

In this section we discuss the obtained results by combining each classifier with every text representation model. To compute the prediction accuracy of a classifier, we used the `metrics` class from `Scikit-learn` library. Then we represented these results as a ROC Curve. The accuracy is calculated based on the test data we had when we split the dataset to train data and test data. The items labels in the test data were removed and stored in another separate variable. We pass the test data to the classifier to see what predictions it results. Then we compare the classifier predictions with the labels we have removed from the test data. The accuracy score is the percentage of the true predictions.

We have the following abbreviations to represent the results of the classifiers in the tables of accuracy and ROC curves: 1. Bag-of-Words: BoW 2. Bi-gram: bigram 3. Term Frequency-Inverse Document Frequency: TF-IDF 4. Multinomial Naive Bayes: MN 5. Linear Support Vector Classifier: LSVC

Content Detection Results We used a dataset with 6335 news articles and performed random split on this dataset into two parts: *training data* and *testing data*. Training data consists of 66.6% of the data and testing data consists of 33.3% of the data.

From Table 3.5, we observe that the LSVC classifier performed well with the TF-IDF model; at the same time the MN gave the worst result with TF-IDF. This is because the MN usually requires integer feature counts to predict with a higher

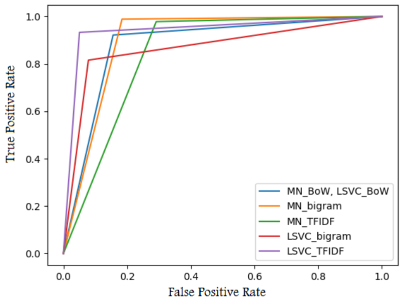


Figure 3.6: ROC Curve Representation for Content Detection

	BoW	TF-IDF	Bigram
MN	0.957	0.956	0.849
LSVC	0.947	0.956	0.845

Table 3.6: Title Detection Accuracy Scores

accuracy. Also, the results show that representing documents through weighted terms vectors (TF-IDF vectors) is more efficient in the case of linear classification. This is because the linear classifiers do not necessary require integer feature counts.

Both classifiers gave the same accuracy score with the BoW model. On the other hand, the MN classifier gave a better score accuracy in case of Bi-gram model. This depends on the the Bi-gram term-document matrix, which has pairs of tokens frequencies as features. This means that using the frequencies of pairs to calculate the probabilities gives better results than mapping these frequencies in a linear space; the reason is that having a pair of tokens with a high frequency can indicate a bigger probability for one of the labels than the other.

Title Detection Results From Table 3.6, we observe that the accuracy scores from both classifiers with the models BoW and TF-IDF are the highest. This is because the dataset we used has a large number of titles, hence a large dataset means large number of features. At the same time, titles are short documents which makes it easier to indicate to which class variables they belong to. This is because in short documents, as a title, there is a small number of features. So the frequencies of a word will have more impact on the classification process, in comparison with large documents which have large number of features where the prediction process will be less accurate. On the other hand, accuracy scores obtained using Bi-gram model were the lowest with both classifiers. This is because the Bi-gram feature extraction strategy gives relatively low results in case of short documents. This is because a large number of Bi-gram frequencies will be 0 or 1, so it will be harder to the classifier to create a clear classification pattern.

The dataset we used to detect if a title is Clickbait/nonClickbait had 32000 news titles. We performed randomly split on this dataset into two parts. Training data consists of 66.6% of the data and testing data consists of 33.3% of the data.

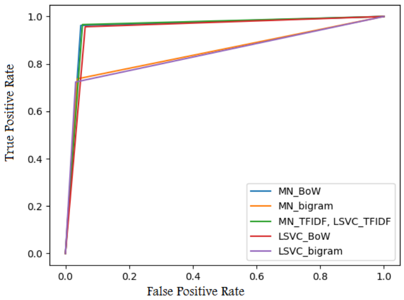


Figure 3.7: ROC Curve representation for title detection

3.4.2 Architecturing Binarized Neural Networks for Traffic Sign Recognition

Binarized Neural Networks

A BNN [67] is a feedforward network where weights and activations are mainly binary. [87] describes BNNs as sequential composition of blocks, each block consisting of linear and non-linear transformations. One could distinguish between *internal* and *output blocks*.

There are typically several *internal blocks*. The layers of the blocks are chosen in such a way that the resulting architecture fulfills the requirements of accuracy, model size, number of parameters, for example. Typical layers in an internal block are: *1)* linear transformation (LIN), *2)* binarization (BIN), *3)* max pooling (MP), *4)* batch normalization (BN). A linear transformation of the input vector can be based on a fully connected layer or a convolutional layer. In our case is a convolution layer since our experiments have shown that a fully connected layer can not synthesize well the features of traffic signs, therefore, the accuracy is low. The linear transformation is followed either by a binarization or a max pooling operation. Max pooling helps in reducing the number of parameters. One can swap binarization with max pooling, the result would be the same. We use this sequence as Larq [57], the library we used in our experiments, implements convolution and binarization in the same function. Finally, scaling is performed with a batch normalization operation [70].

There is *one output block* which produces the predictions for a given image. It consists of a dense layer that maps its input to a vector of integers, one for each output label class. It is followed by function which outputs the index of the largest entry in this vector as the predicted label.

We make the observation that, if the MP and BN layers are omitted, then the input and output of the internal blocks are binary, in which case, also the input to the output block. The input of the first block is never binarized as it drops down drastically the accuracy.

Datasets and Experimental Setting

We use GTSRB [3] for training and testing purposes of various architectures of BNNs. These architectures were also tested with the Belgian data set [1] and the Chinese [2].

GTSRB is a multi-class, single-image dataset. The dataset consists of images of German road signs in 43 classes, ranging in size from 25 × 25 to 243 × 225, and not all of them are square. Each class comprises 210 to 2250 images including prohibitory signs, danger signs, and mandatory signs. The training folder contains 39209 images; the remaining 12630 images are selected as the testing set. For training and validation the ratio 80:20 was applied to the images in the train dataset. GTSRB is a challenging dataset even for humans, due to perspective change, shade, color degradation, lighting conditions, just to name a few.

The *Belgium Traffic Signs* dataset is divided into two folders, training and testing, comprising in total 7095 images of 62 classes out of which only 23 match the ones from GTSRB. Testing folder contains few images for each remaining classes, hence, we have used only the images from the training folder which are 4533 in total. The *Chinese Traffic Signs* dataset contains 5998 traffic sign images for testing of 58 classes out of which only 15 match the ones from GTSRB. For our experiments, we performed the following pre-processing steps on the Belgium and Chinese datasets, otherwise the accuracy of the trained model would be very low: *1)* we relabeled the classes from the Belgium, respectively Chinese, datasets such that their common classes with GTSRB have the same label, and *2)* we eliminated the classes not appearing in GTSRB.

In the end, for testing, we have used 1818 images from the Belgium dataset and 1590 from the Chinese dataset.

For this study, the following points are taken into consideration.

1. Training of network is done on Intel Iris Plus Graphics 650 GPU using Keras v2.10.0, Tensorflow v2.10.0 and Larq v0.12.2.

2. From the open-source Python library Larq [57], we used the function `QuantConv2D` in order to binarize the convolutional layers except the first. Subsequently, we denote it by QConv. The `bias` is set to `False` as we observed that does not influence negatively the accuracy but it reduces the number of parameters.

3. Input shape is fixed either to 30 × 30, 48 × 48, or 64 × 64 (px × px). Due to lack of space, most of the experimental results included are for 30 × 30, however all the results are available at `https://github.com/apostovan21/BinarizedNeuralNetwork`.

4. Unless otherwise stated, the number of epochs used in training is 30.

5. Throughout the paper, for max pooling, the kernel is fixed to non-overlapping 2 × 2 dimension.

6. Accuracy is measured with variation in the number of layers, kernel size, the number of filters and of neurons of the internal dense layer. Various combination of the following values considered are: *(a)* Number of blocks: $2, 3, 4$; *(b)* Kernel size: $2, 3, 5$; *(c)* Number of filters: $16, 32, 64, 128, 256$; *(d)* Number of neurons of the internal dense layer: $0, 64, 128, 256, 512, 1024$.

7. ADAM is chosen as the default optimizer for this study. For initial training of deep learning networks, ADAM is the best overall choice [96].

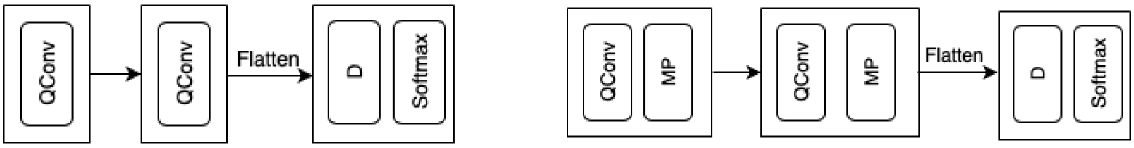
Following section discusses the systematic progress of the study.

Proposed Methodology

We recall that the goal of our work is to obtain a set of architectures for BNNs with high accuracy but at the same time with small number of parameters for the scalability of the formal verification. At this aim, we proceed in two steps. First, we propose two simple two internal blocks XNOR architectures[19] (Section 3.4.2). We train them on a set of images from GTSRB dataset and test them on similar images from the same dataset. We learned that MP reduces drastically the accuracy while the composition of a convolutional and binary layers (QConv) learns well the features of traffic signs images. In Section 3.4.2, we restore the accuracy lost by adding a BN layer after the MP one. At the same time, we try to increase the accuracy of the architecture composed by blocks of the QConv layer only by adding a BN layer after it.

Second, based on the learnings from Sections 3.4.2 and 3.4.2, as well as on the fact that a higher number of internal layers typically increases the accuracy, we propose several architectures (Section 3.4.2). Notable are those with accuracy greater than 90% for GTSRB and an average greater than 80% considering also the Belgian and Chinese datasets, and for which the number of parameters varies from 100k to 2M.

XNOR Architectures We consider the two XNOR architectures from Figure 3.8. Each is composed of two internal blocks and an output dense (fully connected) layer. Note that, these architectures have only binary parameters. For the GTSRB, the results are in Table 3.7. One could observe that a simple XNOR architecture gives accuracy of at least 70% as long as MP layers are not present but the number of parameters and the model size are high. We can conclude that QConv synthesizes the features well. However, MP layers reduce the accuracy tremendously.



(a) XNOR(QConv) architecture

(b) XNOR(QConv,MP) architecture

Figure 3.8: XNOR architectures

Binarized Neural Architectures

Two internal blocks As of Table 3.7, the number of parameters for an architecture with MP layers is at least 15 times less than in a one without, while the size of the binarized models is approx. 30 times less than the 32 bits equivalent. Hence, to benefit from these two sweet spots, we propose a new architecture (see Figure 3.9b) which adds a BN layer in the second block of the XNOR architecture

[19] An XNOR architecture [92] is a deep neural network where both the weights and the inputs to the convolutional and fully connected layers are approximated with binary values.

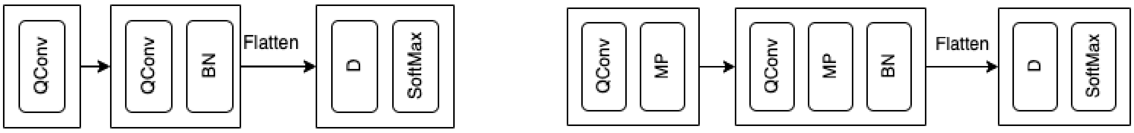
Table 3.7: XNOR(QConv) and XNOR(QConv, MP) architectures. Image size: 30px × 30px. Dataset for train and test: GTSRB.

Model description	**Acc**	**#Binary Params**	**Model Size (in KiB)**	
			Binary	**Float-32**
QConv(32, 3×3), QConv(64, 2×2), D(43)	77.91	2015264	246.5	7874.56
QConv(32, 3×3), MP(2×2), QConv(64, 2×2), MP(2×2), D(43)	5.46	108128	13.2	422.38
QConv(64, 3×3), QConv(128, 2×2), D(43)	70.05	4046912	495.01	15810.56

QConv(64, 3×3), MP(2×2),
QConv(128, 2×2), MP(2×2)
D(43)
	10.98	232640	28.4	908.75

QConv(16, 3×3), QConv(32, 2×2), D(43)	81.54	1005584	122.75	3932.16
QConv(16, 3×3), MP(2×2), QConv(32, 2×2), MP(2×2), D(43)	1.42	52016	6.35	203.19

from Figure 3.8b. The increase in accuracy is considerable (see Table 3.8)[20]. However, a BN layer following a binarized convolution (see Figure 3.9a) typically leads to a decrease in accuracy (see Table 3.9). The BN layer introduces few real parameters in the model as well as a slight increase in the model size. This is because only one BN layer was added. Note that the architectures from Figure 3.9 are not XNOR architectures.



(a) XNOR(QConv) modified

(b) XNOR(QConv, MP) enhanced

Figure 3.9: BNNs architectures which are not XNOR

Table 3.8: XNOR(QConv, MP) enhanced. Image size: 30px ×30px. Dataset for train and test: GTSRB.

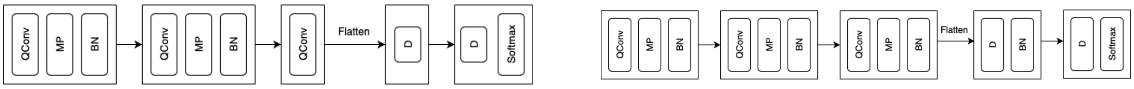
Model description	**Acc**	**#Params**			**Model Size (in KiB)**	
		Binary	**Real**	**Total**	**Binary**	**Float-32**
QConv(32, 3×3), MP(2×2), QConv(64, 2×2), MP(2×2), BN, D(43)	50.87	108128	128	108256	13.7	422.88
QConv(64, 3×3), MP(2×2), QConv(128, 2×2), MP(2×2), BN, D(43)	36.96	232640	256	232896	29.4	909.75
QConv(16, 3×3), MP(2×2), QConv(32, 2×2), MP(2×2), BN, D(43)	39.55	52016	64	52080	6.6	203.44

Several Internal Blocks Based on the results obtained in Sections 3.4.2 and 3.4.2, firstly, we trained an architecture where each internal block contains

[20] A BN layer following MP is also obtained by composing two blocks of XNOR-Net proposed by [92].

Table 3.9: XNOR(QConv) modified. Image size: 30px × 30px. Dataset for train and test: GTSRB.

Model description	Acc	#Params			Model Size (in KiB)	
		Binary	Real	Total	Binary	Float-32
QConv(32, 3×3), QConv(64, 2×2), BN, D(43)	82.01	2015264	128	2015392	246.5	7874.56
QConv(64, 3×3), QConv(128, 2×2), BN, D(43)	69.12	4046912	256	4047168	495.01	15810.56
QConv(16, 3×3), QConv(32, 2×2), BN, D(43)	73.11	1005584	64	1005648	123	3932.16



(a) 4-blocks Binarized Neural Architecture

(b) Accuracy-efficient Binarized Neural Architectures

Figure 3.10: Binarized Neural Architectures

a BN layer only after the MP (see Figure 3.10a). This is based on the results from Tables 3.8 (the BN layer is crucial after MP for accuracy) and 3.9 (BN layer after QConv degrades the accuracy). There is an additional internal dense layer for which the number of neurons varies in the set $\{64, 128, 256, 512, 1028\}$. The results are in Table 3.10. One could observe that the conclusions drawn from the 2 blocks architecture do not persist. Hence, motivated also by [67] we propose the architecture from Figure 3.10b.

Table 3.10: Results for the architecture from the column Model Description. Image size: 30px ×30px. Dataset for train and test: GTSRB.

Model Description	#Neur	#Ep	Acc	#Params			Model size (in KiB)	
				Binary	Real	Total	Binary	Float-32
QConv(32, 5x5), MP(2x2), BN, QConv(64, 5x5), MP(2x2), BN, QConv(64, 3x3), D(#Neur), D(43)	0	30	41.17	101472	192	101664	13.14	397.12
		100	52.17					
	64	30	4.98	109600	192	109792	14.13	428.88
		100	5.7					
	128	30	7.03	128736	192	128928	16.46	503.62
		100	5.70					
	256	30	12.43	167008	192	167200	21.14	653.12
		100	8.48					
	512	30	19.82	243552	192	243744	30.48	952.12
		100	32.13					
	1024	30	46.05	396640	192	396832	49.17	1546.24
		100	50.91					

Experimental results and discussion

The best accuracy for GTSRB and Belgium datasets is $96,45$ and $88,17$, respectively, and was obtained for the architecture from Figure 3.11, with input size 64×64 (see Table 3.11). The number of parameters is almost 2M and the model size $225,67$ KiB (for the binary model) and $6932,48$ KiB (for the Float-32 equivalent). There is no surprise the same architecture gave the best results for GTSRB and Belgium since they belong to the European area. The best accuracy for Chinese dataset ($83,9\%$) is obtained by another architecture, namely from Figure 3.12,

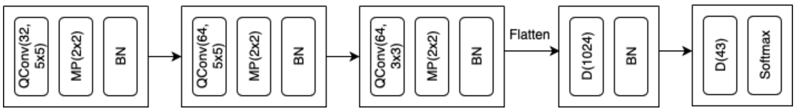


Figure 3.11: Accuracy Efficient Architecture for GTSRB and Belgium dataset

Table 3.11: Results for the architecture from Figure 3.11. Dataset for train: GTSRB.

Input size	**#Neur**	**Accuracy**			**#Params**			**Model Size (in KiB)**	
		German	**China**	**Belgium**	**Binary**	**Real**	**Total**	**Binary**	**Float-32**
64px × 64px	0	93.83	77.86	79.75	159264	320	159584	20.69	623.38
	64	94.43	75.09	82.39	195616	448	196064	25.63	765.88
	128	95.42	74.71	83.44	300768	576	301344	38.96	1177.60
	256	94.75	80.37	81.40	511072	832	511904	65.64	1996.80
	512	95.65	78.49	85.64	931680	1344	933024	118.98	3645.44
	1024	**96.45**	**81.50**	**88.17**	1772896	2368	1775264	225.67	6932.48

with input size 48×48 (see Table 3.12). This architecture is more efficient from the point of view of computationally limited devices and formal verification having 900k parameters and $113,64$ KiB (for the binary model) and $3532,8$ KiB (for the Float-32 equivalent). Also, the second architecture gave the best average accuracy and the decrease in accuracy for GTSRB and Belgium is small, namely $1,17\%$ and $0,39\%$, respectively.

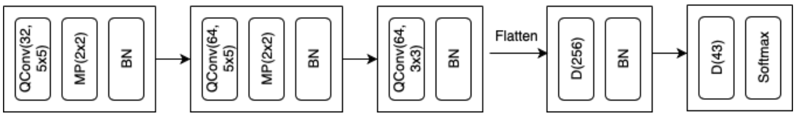


Figure 3.12: Accuracy Efficient Architecture for Chinese dataset

If we investigate both architectures based on confusion matrix results, for GTSRB we observe that the model failed to predict, for example, the *End of speed limit 80* and *Bicycle Crossing*. The first was confused the most with *Speed limit (80km/h)*, the second with *Children crossing*. One reason for the first confusion could be that *End of speed limit (80 km/h)* might be considered the occluded version of *Speed limit (80km/h)*.

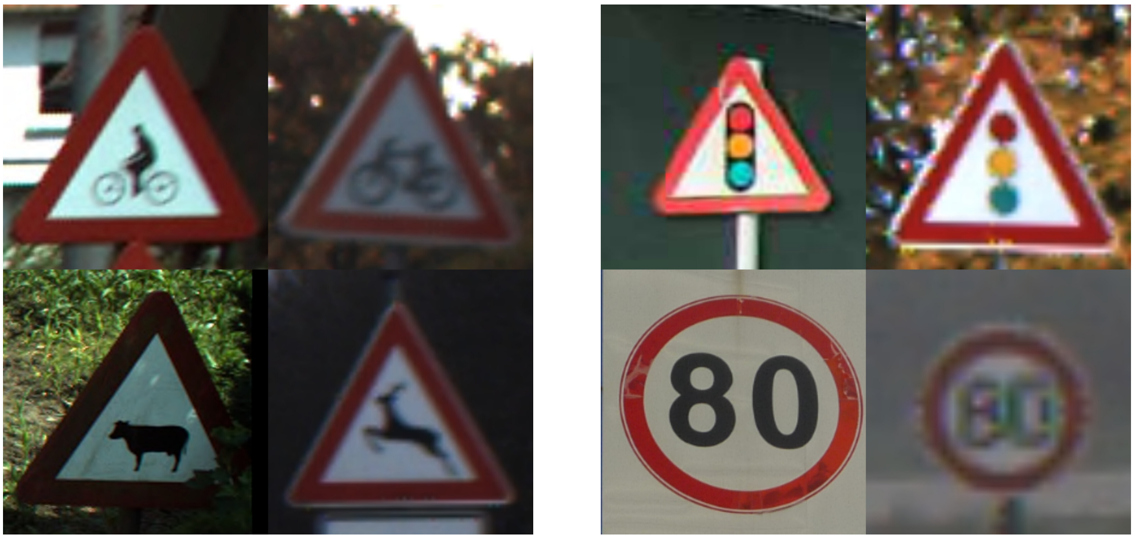
For Belgium test set, the worst results were obtained, for example, for *Bicycle crossing* and *Wild animals crossing* because the images differ a lot from the images on GTSRB training set (see Figure 3.13a). Another bad prediction is for *Double Curve* which was equally confused with *Slippery road* and *Children crossing*.

In the Chinese test set, the *Traffic signals* failed to be predicted at all by the model proposed by us and was assimilated with the *General Caution* class from the GTSRB, however *General Caution* is not a class in the Chinese test set (see Figure 3.13b, top). Another bad prediction is for *Speed limit (80km/h)* which was equally confused with *Speed limit (30km/h), Speed limit (50km/h)* and *Speed limit (60km/h)* but not with *Speed limit (70km/h)*. One reason could be the quality of the training images compared to the test ones (see Figure 3.13b, bottom).

In conclusion, there are few cases when the prediction failures can be explained, however the need for formal verification guarantees of the results is urgent which we will be performed as future work.

Table 3.12: Results for the architecture from Figure 3.12. Dataset for train: GTSRB.

Input size	#Neur	Accuracy			#Params			Model Size (in KiB)	
		German	China	Belgium	Binary	Real	Total	Binary	Float-32
48px × 48px	0	94.67	82.13	83.16	225312	320	225632	28.75	881.38
	64	94.56	82.38	85.75	293920	448	294368	37.63	1146.88
	128	95.02	81.50	87.45	497376	576	497952	62.96	1945.60
	256	**95.28**	**83.90**	**87.78**	904288	832	905120	113.64	3532.80
	512	95.90	76.22	87.34	1718112	1344	1719456	214.98	6717.44
	1024	95.37	81.76	86.74	3345760	2368	3348128	417.67	13076.48



(a) Difference between Belgium (left) and GRSRB (right) dataset

(b) Difference between Chinese (left) and GRSRB (right) dataset

Figure 3.13: Differences between traffic sign in the datasets

3.4.3 Conclusions

In this section, we presented classification techniques for fake news detection and, respectively, traffic signs recognition.

In today's interconnected world, the pervasive issues of fake news and disinformation have taken center stage. The rapid advancement of technology and the ease of communication have led to the unverified dissemination of information, prompting researchers to actively seek remedies. In [10], we addressed these challenges by proposing an algorithm that evaluates the credibility of news articles based on their unique attributes. By integrating a blend of classification methods and text models, the algorithm exhibits commendable performance, as reflected in the accuracy results.

Looking ahead, our focus will involve a nuanced exploration of the symbiotic relationship between feature extraction techniques and classifiers. This endeavor aims to pinpoint the optimal text representation model that seamlessly aligns with the selected classifier, thereby enhancing the system's overall efficacy. Furthermore, our trajectory involves the deployment of a more intricate algorithm, potentially harnessing data mining strategies tailored for expansive datasets. Encompassing diverse news article types and a rich array of class variables (labels), these datasets offer the promise of elevating accuracy scores and presenting a comprehensive solution to the challenges of fake news and disinformation in today's information landscape.

Moving in the direction of autonomous driving, it is well-know that traffic signs play a pivotal role in ensuring road safety and efficient traffic management, making them an indispensable component of autonomous driving vision systems. While deep learning, particularly convolutional neural networks (CNNs), is renowned for its traffic sign classification prowess (achieving up to 99.46% accuracy), the potential of binarized neural networks (BNNs) remains largely unexplored. BNNs offer benefits such as reduced model size and simplified convolution operations, making them promising candidates for energy-constrained autonomous driving environments. In [91], we introduced a novel approach to designing BNN architectures with high accuracy, involving an in-depth examination of constituent layers (binarized convolutional layers, max pooling, batch normalization, and fully connected layers) , explored in various configurations with diverse kernel sizes, filter counts, and neuron quantities.

Future work could follow two paths:

1. machine learning: *(1)* investigate the accuracy by using other pooling operators (e.g. average pooling), *(2)* consider only fully connected layers and ReLU activations but high number of neuraons *(3)* interpret the accuracy and loss of the trained architectures

2. (robustness) verification: we submitted the best accuracy models we trained together with local robustness properties to be verified to the VNN-COMP 2023 in order to find out if the state-of-the-art tools can handle them. 4 out of 7 tools produced results. Our current emphasis is on investigating the potential for solving more instances by extending the time limit (in the competition this was set at 8 minutes). Additionally, we are keen to comprehend the factors contributing to incorrect answers from the tools on specific benchmark tasks. These are short term goals.

   On medium-long term, we plan to explore a different path on checking robustness of a machine learning classifier using formal verification, not at pixel level, as most works consider now, but at object level. This means checking if the same object is represented in two images.

3.5 Computer Science Education

This section is about the impact of transferring learning into the workplace of an Informatics teacher (the author of this thesis) seeking to promote student-centered learning (SCL) within a new discipline in her teaching portfolio (i.e., *Software Engineering*).

The starting point of this work was the motivation that, over time, instructional training activities for academics that promote student-centered learning (SCL) increased. However, few things are known about the extent to which academics' learning is transferred into the daily teaching practice.

For studying the impact of transferring learning into the workplace a quasi-experimental design with pre- and post-test was employed. Self-reported data were collected as follows: from the experimental group, there were 52 students (28.8% female) at the pre-test, and 29 students (37.9% female) at the pre-test, while from the control group, data were collected from 26 students (34.6% female) at the pretest and 19 students (47.3% female) at the post-test. Independent t-test analysis showed that the SCL initiative had only a positive impact on student learning approaches and teaching quality as perceived by students. Concerning students' learning approaches, the SCL initiative had no effect. Several interpretations and perspectives of the current study are discussed.

This result appeared in [53] and it is presented in Section 3.5.1.

3.5.1 Transferring Learning into the Workplace: Evaluating a Student-Centered Learning Approach through Computer Science Students' Lens

Teaching quality enhancement to improve student learning is still an ongoing concern for most higher education institutions worldwide. Specifically, in Europe, mainly because it influences student achievement, since the Bologna Process, student-centered learning (SCL) became the primary instructional approach [103]. SCL, among other aspects, focuses on the student's needs (e.g., the curriculum and courses are more flexible, the learning process is more interactive), aiming to facilitate students' adoption of deep learning approaches [73]. Therefore, many resources were invested to improve staff development initiatives, develop efficient *instructional development programs* (IDPs), assess and enhance teaching quality, offer incentives for teaching excellence, etc. [102]. Consequently, there is a massive requirement for quality evidence of IDPs' or staff development impact on daily teaching practices [40].

Some studies treated IDPs and the staff development concept as similar concepts. Hence, they have several related terms: academic development, instructional training, educational development, faculty, or professional development. In this study, IDPs and staff development initiatives are treated as correlated but different constructs. Thus, we will refer to IDPs as any initiative precisely planned to enhance academics' teaching (i.e., in their role as a teacher) to support student learning [104]. On the other hand, we will refer to staff development initiatives as a sum of informal (e.g., exchange of ideas among teachers) and formal (e.g., workshops) learning experiences of the teacher [55]. In staff development initiatives, academics have to translate their acquired competencies (e.g., knowledge, skills, attitudes) into changes in their thinking and educational behavior. Therefore, in the present study, we will consider [14] definition to define the transfer of the acquired competencies (e.g., learning) to the workplace (i.e., in the classroom) due to IDPs or staff development initiatives.

However, mainly because of the limited resources, in the regular practice, IDPs' and staff development initiatives impact is generally assessed at one level (e.g., teachers' attitude or knowledge, students' learning approach, or perception of teaching quality) [101, 103]. The latest reviews in the impact assessment of staff development recommended that the impact of IDPs should be measured on several levels of outcomes [75] by well-designed studies (e.g., with at least a quasi-experimental or a longitudinal approach) [69, 104]. Even though not without limitations, the present article aims to bring more evidence regarding a specific SCL teaching initiative (i.e., as a result of attendance of an IDP and of a staff development initiative) of learning transfer to the workplace. *Using a quasi-experimental design with pre- and* *post-test, the current endeavor evaluated the impact at two different levels: students' perceptions of teaching quality and students' approaches to learning.*

Students' perceptions of teaching quality or student evaluation of teaching (SETs) represent one of the most voluminous literature research works in the applied psychology field [63]. Paper [79] suggests that teaching evaluation in higher education institutions is important for two main reasons. First, through this evaluation, one can improve teachers' performance by offering them feedback and designing IDPs directed on the identified training needs. Second, one can use the results from SETs in administrative decisions like promotion, rewards, and external accountability. Regarding the impact of IDPs or staff development initiative on students' perceptions of teaching quality, most of the studies presented mixed results (e.g., positive impact [61, 81]; negative impact [102]. Therefore, for a clearer picture, more studies are needed.

Nowadays, successful learning and studying in higher education is most often associated with students' deep approaches to Learning [11]. A deep learning approach is characterized by significant engagement in the learning process, independent thinking, analytic skills, and understanding of the subject matter [12]. On the other side, there is the undesired surface learning approach. Its short-term benefits involve memorizing the subject matter without understanding its utility or implications [12]. Nevertheless, helping students transition towards a deep approach to learning is not an easy task [13]. Few studies showed that students attending classes held by teachers who completed an IDP increased their deep learning approaches compared to the students from the control group [61]. Nonetheless, studies investigating the students' level changes due to their teachers' participation in an IDP are scarce [69].

Design and aim of the study and hypotheses

The present study used a quasi-experimental design including a pre-test and post-test to assess the transfer into the workplace of an SCL initiative. More precisely, the present study evaluated the degree of the transfer into the workplace of an SCL approach into the context of teaching the Software Engineering subject for bachelor Computer Science students. Therefore, we evaluated the changes in students' perception of teaching quality and students' approaches to learning. Specifically, we advanced three research questions:

Q1. Is there any progress in students' approaches to learning from the experimental group due to the learning transfer into the workplace of the SCL initiative implemented by their teacher?

Q2. Are there any statistically significant differences between the experimental and control group students regarding their approaches to learning?

Q3. Is the teacher's teaching that implemented the SCL initiative perceived as better by her students than students' perception of the teaching of her counterpart in the control group at the end of the semester?

Before introducing the method and results of the study, we present an outline of the learning transfer into the workplace of an SCL initiative in question.

Research context

West University of Timisoara, Romania, organizes and encourages participation at several IDPs that promote SCL. The first author of this paper participated in an IDP and a staff development initiative. The IDP (i.e., University didactics and psychopedagogy) was attended between February and March 2020, having the following structure: 5 disciplines, cumulating 150 hours, of which 40 hours theoretical courses (10 hours/discipline, within four disciplines) and 80 hours of practical applications (20 hours/discipline, in 4 disciplines - The Management of the Students Groups, Elaboration of the Didactic Materials, Modern Methods of Education, Curricular Design) and 30 hours of practical applications in the fifth discipline (i.e., Feedback and Didactic Counseling). The primary purpose of this IDP was to improve the level of competencies of the university teaching staff regarding the development of educational offers with innovative and student-centered instructive-educational content and approaches. Regarding its gains, besides belonging to a learning community, at the end of the IDP, each participant has a complete curricular package (e.g., syllabus, teaching strategies, activity plans, assessment tools, etc.) for a discipline they teach in the current practice.

After graduating from the early mentioned IDP, the first author applied and won one of the twenty didactic incentives (i.e., inside the competition Didactic Grants) supported by the university to further implement the SCL approach in the classroom. The staff development initiative (i.e., Didactic Grants Competition) involved a training schedule similar to the University didactics and psychopedagogy IDP (but much shorter and less complex), plus several other informal activities (e.g., informal counseling meetings via Google Meet). As a graduate of the IDP, the first author of this paper did not have to repeat the training activities. However, she was supposed to complete all the other outputs of the Didactic Grants Competition initiative (e.g., design three activity plans and implement at least one of them; record a teaching activity, etc.). Thereby, in the summer semester of 2021, she applied the acquired competencies in the IDP and the staff development initiative to the lecture and laboratory of Software Engineering, a new subject in her teaching portfolio.

Learning transfer into the workplace of the SCL initiative

Regarding the adopted SCL approach, we mention that the discipline taught to the experimental group was Software Engineering, second year, undergraduate level. Introductory topics in this field were presented based on the books [106] for the course, respectively [99] for the laboratory. The chosen topics were such that they prepare the students to understand the basic notions when working for software companies and writing their Bachelor thesis at the end of the third year. At this aim, the lecture was structured as follows:

1. *Software Management:* The Software Life Cycle and variants (advantages and disadvantages): The Waterfall Model, Agile Methods, Prototyping, Incremental Development, Rapid Application Development, and DSDM, Extreme Programming; The Rational Unified Process (RUP); Intermezzo: Maintenance or Evolution; Software Product Lines; Process Modelling;

2. *The Software Life Cycle:* Requirements Engineering; Modelling; Architecture; Design.

The laboratory focused on UML modelling, emphasizing the following topics: Use-case diagram, Class diagram, State-machine diagram, Sequence diagram, and Activity diagram. All the semester, the classes were held online due to the Covid19 breakthrough. During the semester, the most challenging was to keep the students focused and engaged. At this aim:

1. we designed the course and laboratory to be very interactive, and

2. the knowledge assessment was continuous during the whole semester.

In the Romanian university system, each course and laboratory last 90 minutes. We did our best to divide this time as follows:

1. Clearly define the objectives of the current course and laboratory and relate them with the learning results of the discipline and with previous and future ones.

2. A session in which the teacher presented new material was of maximum 20 minutes and was always followed by a practical session (individual or in teams) and a reflection of what was taught.

3. We tried that each course/laboratory was concluded with a summary of what was studied. This summary was presented by a student randomly chosen from the group (to keep students' attention).

The knowledge assessment was done continuously during the semester. It was composed of:

1. quizzes during the lecture,

2. examination in the exam session composed of short questions and synthesis subjects to prove that the students deeply understood the topics, and

3. team project for the laboratory.

These three components summed up 10 points, which is the maximum grade in the Romanian grading system. There was also the possibility to choose an individual project on actual topics of research in software engineering. This, together with excellent activity during the semester (at least 9 points for quizzes and team project), would have given the students the possibility to have the maximum grade without taking the final examination in the exam session.

Method

Participant Characteristics Students in both experimental and control groups were similar in terms of faculty (i.e., Faculty of Mathematics and Informatics), specialization (i.e., Applied Informatics), year of studies (2nd year), degree (i.e., bachelor's degree), the academic status of their teacher (i.e., University lecturers), and teaching experience of their teacher (i.e., > 5 years). As compared to the teacher who was the one responsible for the learning transfer into the workplace of the SCL initiative, the counterpart teacher (i.e., the teacher from the control group) did not follow any IDP or staff development initiative on SCL. The distribution of students' mean age, gender, class size and type of activity, and area of residence are presented in Table 3.13.

Measures For the data collection, we used two instruments *Revised Two-Factor Study Process Questionnaire* (R-SPQ-2F [24]) and *Exemplary Teacher Course Questionnaire* (ETCQ [74]), both being previously used on the Romanian population [100, 68]. The R-SPQ-2F measures students' preferences for study strategies [12]. The R-SPQ-2F has 20 items, assessing two learning approaches, namely the deep and surface learning approach. Each dimension of the R-SPQ-2F is divided into two corresponding subscales (i.e., motives and strategies). The items gather answers through a 5-point Liker scale (i.e., from 1 = never/only rarely true of me to 5 = always/almost always true of me). In terms of factorial structure of R-SPQ-2F, we used the 2-factor one as it proved to be superior on the Romanian population [100]. The deep learning approach scale measures students' motives and strategies described by intrinsic motivation and maximization of their understanding of the discipline. On the other hand, the surface learning approach describes motives and strategies related to extrinsic motivation involving memorizing the course without understanding its implications or utility. We chose the ETCQ mainly because of its validity, reliability, and diagnostic power [74, p.352]. The ETCQ has 49 items, assessing nine dimensions (Table 3.17) of the teaching process in the classroom environment as perceived by students. The responses to each of the nine dimensions of ETCQ were gathered with a 5-point Likert scale (i.e., ranging from 1 = strongly disagree to 5 = strongly agree). Reliability coefficients for the two scales of the R-SPQ-2F and the nine scales of ETCQ for both the control and experimental group at the two collection data time points (i.e., pre- and post-test) are presented in Table 3.14.

Table 3.13: Demographic characteristics of the student sample.

Students' characteristic	Pre-test		Post-test	
	Experimental group	Control group	Experimental group	Control group
Mean age	20.31	20.73	21.14	20.74
Gender				
Female	15	9	11	9
Male	33	16	16	10
Not mentioned	4	1	2	0
Class size & type of activity				
≤ 30 students (seminary)	-	26	-	19
>30 but ≤ 60 students (lecture)	52	-	29	-

Data collection We used the aforementioned instruments and assembled a quantitative pre-test (first week of the semester) and a post-test (last week). Participation in the current study was voluntary for all students, and all answers were anonymous. Before completing the questionnaires, a researcher read one standard procedure to fill in the questionnaire. Each student had an anonymous research code to help the research team match their answers from pre-test to post-test. However, excepting the repeating students (i.e., which were too few) the rest of the students that participated in the post-test, according to the research code, were not

Table 3.14: ETCQ and R-SPQ-2F α Cronbach's indices for the experimental and control group at the pre-test and post-test.

Questionnaire / scale	Moment	No. of items	Alpha Cronbach α	
			Experimental group	Control group
Revised Two-Factor Study Process Questionnaire [24]				
Deep Learning Approach	pre-test	10	.810	.687
	post-test		.853	.837
Surface Learning Approach	pre-test	10	.797	.803
	post-test		.847	.837
Exemplary Teacher Course Questionnaire [74]				
Understanding Fundamental Concepts	post-test	5	.826	.842
Relevance	post-test	5	.743	.859
Challenging Beliefs	post-test	6	.885	.823
Active Learning	post-test	5	.702	.873
Teacher-Student Relationships	post-test	5	.797	.886
Motivation	post-test	6	.868	.883
Organization	post-test	7	.944	.942
Flexibility	post-test	5	.868	.948
Assignments	post-test	5	.760	.655

the same as those in the pre-test. In the case of R-SPQ-2F, data were collected for both pre-test and post-test. Students were instructed to report their general study approaches about the study program they followed (i.e., Applied informatics) at the pre-test. On the other hand, at the end of the semester (i.e., at the post-test), students were asked to report their specific learning approaches in the case of the followed discipline (i.e., Software Engineering in the case of the experimental group and Databases Administration in the case of the control group). As students cannot accurately refer to the teacher's behavior with whom they did not study before, in the case of ETCQ, data were gathered only in the post-test moment.

Data analysis First, we assessed Cronbach's α for each experimental and control group subscale for pre-test and post-test moments. All the obtained values for Cronbach's α indicated acceptable reliability, with almost all scales having good or very good reliability ($\alpha > .80$) (Table 3.13). Second, given the design of our study (i.e., same teachers and courses were considered for both pre-test and post-test moments) and that only a few students answered the questionnaires in both evaluation moments, we could not perform a paired sample t-test. Therefore, to determine that involved the inspection of normality and homogeneity of variance assumptions: normal plots, stem and leaf plots, and the calculation of skewness and kurtosis were used to verify the normality of the data distribution, while the Levene statistics were calculated to test the equality of group variances. All the preliminary assumptions for the analysis were met (i.e., we have a continuous dependent variable; the independent variable has two categorical, independent groups; the observations are independent; there are no significant outliers; the data distribution is normal) for most of the ETCQ dimensions, excepting the active learning dimension where the equal variances assumption was violated. Therefore, in the case of the active learning dimension, we followed the recommendation of (Howell, 2012), and we performed the Welch t-test (i.e., the nonparametric version of interdependent t-test), while for the rest of the R-SPQ-2F and ETCQ dimensions, we used the interdependent t-test.

Table 3.15: Student's approaches to study at the beginning, respectively at the end of the semester for students in the experimental group.

Group		Deep Learning Approach			Surface Learning Approach		
		Mean score	SD	N	Mean score	SD	N
Experimental	Before	2.72	0.66	52	2.44	0.68	52
	After	2.66	0.72	29	2.49	0.78	29
	Change	-.06			.05		
	t	.417			-.295		
	p	.678			.769		

Results

Research Question 1. Regarding the scores of the students in the experimental group, Table 3.15 presents their approaches to studying at the beginning and the end of the semester, respectively. No statistically significant improvements can be discerned. However, there is an elusive decrease in the deep learning approaches from the pre-test to the post-test moment (i.e., change = -.06 with Mpre-test = 2.72, Mpost-test = 2.66), respectively an elusive increase (i.e., change = +.05 with Mpre-test = 2.44, Mpost-test = 2.49) in students' surface learning approaches.

Research Question 2. As of Table 3.16, there are no statistically significant differences between the two groups of students regarding their approaches to learning none of the assessment moments.

Table 3.16: Comparison between experimental and control group before and after the end of semester regarding student's learning approaches.

Variables	Moment	Group						t	df	p
		Experimental			Control					
		N	M	SD	N	M	SD			
Deep Learning Approach	pre-test	52	2.72	0.66	26	2.74	0.54	-0.089	76	0.93
	post-test	29	2.66	0.72	19	2.89	0.66	-1.140	46	0.26
Surface Learning Approach	pre-test	52	2.44	0.68	26	2.60	0.65	-0.969	76	0.34
	post-test	29	2.49	0.78	19	2.79	0.73	-1.327	46	0.191

Research Question 3. Independent t-test analysis for the differences in students' perception of the teaching quality revealed statistically significant differences for only three out of the nine scales of the ETCQ (Table 3.17). First, there is a marginally statistically significant difference regarding the active learning behaviors of the two teachers: the students in the experimental group reported more behaviors of their teacher that encouraged and facilitated their active learning than the students in the control group (t[27.76]= 1.891, p = .069, d Cohen = 0.58). Second, students in the experimental group reported lower scores concerning their relationship with their teacher than students in the control group, which said they had a better relationship with their teacher (t[46]= -2.065, p = .045, d Cohen = 0.61). Third, there is a marginally statistically significant difference regarding the organization of the two courses. Students from the control group perceive their classes to be better organized by their teacher (t[46]= -1.795, p = .079, d Cohen = 0.53).

Table 3.17: ETCQ dimension scores for the experimental in comparison to the control group at the post-test moment.

ETCQ Scale (Group)	N	SD	Mean score	t	df	p	Change	d Cohen
Understand Fundam. Concepts								
Experim Grp	29	3.68	0.75	-1.252	426	0.217	Same	-
Ctrl Grp	19	3.97	0.86					
Relevance								
Experimental Grp	29	3.73	0.71	-0.178	46	0.859	Same	-
Control Grp	19	3.77	0.71					
Challenging Beliefs								
Experimental Grp	29	3.46	0.92	0.475	46	0.637	Same	-
Control Grp	19	3.34	0.70					
Active Learning								
Experimental Grp	29	4.21	0.56	1.891	27.76	0.069*	Better	0.58
Control Grp	19	3.79	0.87					
Teacher-Student Relationships								
Experimental Group	29	3.35	0.81	-2.065	46	0.045*	Worse	0.61
Control Grp	19	3.84	0.79					
Motivation								
Experimental Grp	29	3.41	0.92	-0.839	46	0.406	Same	-
Control Grp	19	3.63	0.81					
Organization								
Experimental Grp	29	3.45	1.01	-1.795	46	0.079*	Worse	0.53
Control Grp	19	3.96	0.90					
Flexibility								
Experimental Grp	29	3.92	0.82	-0.894	46	0.376	Same	-
Control Grp	19	4.15	0.95					
Assignments								
Experimental Grp	29	3.81	0.78	0.115	46	0.909	Same	-
Control Grp	19	3.79	0.61					

Discussion

In the current study, we investigated the impact of learning transfer into the workplace of an Applied informatics higher university teacher by implementing a student-centered learning (SCL) initiative during one semester on a new subject in its portfolio. Hence, we evaluated the SCL initiative's impact on two levels: students' approaches to learning and students' perception of the teaching quality (i.e., which is also a measure for teacher's teaching behaviors).

Regarding the first two research questions of the current investigation, the SCL initiative did not have any impact on students' learning approaches. There were no improvements either on the deep approaches to learning or on the surface learning approaches of the students in the experimental group. Also, there were no differences regarding students' learning approaches between the control and experimental group. At the end of the semester, students in both groups had the same learning approaches in the two disciplines as, in general, in their bachelor study program. Our results differ from several other studies, which found that students of academics that participated in an IDP or a staff development initiative were more likely to adopt deep learning approaches [61]. On the other side, a recent study by [12] concluded that most of the existing studies do not exhibit clear empirical evidence proving that students develop deep approaches to learning during higher education. Moreover, several other studies showed that the deep approach to learning does not necessarily develop during university studies. Students' deep approach to learning during bachelor study years could decline [77] while the surface approach develops [58].

Concerning the third research question of the present study, there were both expected and unexpected results. The SCL initiative had a positive impact only on the active learning dimension out of the nine dimensions of teaching quality perceived by the students. Effect sizes point towards a medium practically exciting impact of the SCL initiative on the scale of Active learning (d Cohen = 0.58). One of the reasons why the Active learning variable is higher for the experimental group could be because the individual quizzes and/or group tasks were constantly assigned during lectures. Also, a group project with different milestones was set, and feedback was given to all the teams in the experimental group. This result is in line with some other studies [61, 81]. For example, [81] found a positive impact of an IDP for debutant academics towards students' ratings, showing a statistically significant increase in the experimental group compared to the control group. On the other hand, students in the control group reported higher scores on their teacher's behaviors regarding the Teacher-Student Relationships and Organization of the course. This result could be explained by the fact that the discipline taught by the teacher who implemented the SCL initiative was new in her teaching portfolio, this being not the case of the counterpart teacher. Another possible reason could be that students in the experimental group perceived the numerous tasks and homework during the semester and their consistent application as too strict. Also, in most disciplines, students are being evaluated mainly in the examination session. Thus, as suggested by other studies, teachers must allocate extra time to successfully implement what is learned during instructional development in daily practice[61, 101]. [90] showed that changing the paradigm to a SCL approach is slow and progressive on the teachers' side. Hence, one semester counting 14 weeks may not be sufficient for visible results. However, several studies which measured the impact of an IDP or staff development initiative reported no, limited, or even negative effects [103, 101].

Limitations and future directions The main limitation of the current endeavor is the low number of students in the two groups and the impossibility of matching all the responses in the two assessment moments. As a consequence, our statistical power is very low. Second, the employed design is quasi-experimental (i.e., lack of randomization). Third, because of limited resources, we assessed the impact of the SCL initiative only through quantitative investigation. Thus, we should be cautious in interpreting present results for the early mentioned reasons and not only. Future studies should consider employing an experimental design (i.e., conducting a randomized controlled trial), quantitative and qualitative measurements (e.g., classroom observations, interviews, etc.), and most importantly, good statistical power. Also, if possible, one should obtain answers from the same students in the pre-test and post-test.

3.5.2 Conclusions

We presented the effect of an SCL initiative through a quasi-experimental design, with a pre-test and post-test assessment. We showed that transferring learning into the workplace of an Applied informatics higher university teacher by implementing student-centered learning (SCL) is perceived as positive by the students. However, creating an active learning environment may not be enough to convince them to change their usual learning approaches. Hence, one should strive to transfer their learning into daily practice to influence student learning positively.

Chapter 4

Scientific and Professional Roadmap

4.1 Envisaged Scientific and Professional Endeavors

There are several directions for upcoming scientific and academic development that will either continue from the current status or become easier once the habilitation title is granted.

Scientifically, I intend to maintain a good rate of producing and disseminating innovative techniques and applicable results at the intersection of formal methods, artificial intelligence, and real-world applications (see Section 4.2). My primary focus will be on publishing in well-known conference venues and highly-rated journals.

I will continue submitting grant proposals to the national agency UEFISCDI. Additionally, I plan to reinvigorate the past collaborations and make new ones in order to be part of European Commission funded projects. My expertise in writing and evaluating such proposals will aid in drafting my personal ones. My proposals will include generating PhD positions and incorporating postdoctoral team members, as I have done with financed projects in the past. Pursuing individual scholarships, such as those from Fulbright, will also be a means to enhance my scientific development at any stage in my career.

Needless to say, I will continue **serving the community** by organizing different kind of scientific events (conferences, workshops, project meetings) or being part of conferences/journals program committees, as reviewer or chair.

Regarding **combined scientific and academic development**, I will strengthen scientific cooperation with potential future researchers early on their studies. As I do in the current project SAGE, I plan to continue weekly meetings for supervised students to discuss research ideas and receive feedback on their work. Furthermore, I aim to encourage PhD students to pursue a joint supervision scheme (e.g. UNITA alliance where West University is part of), allowing them to gain dual experiences by studying both in Romania and abroad. Also, if the topic of the research is relevant to the market, I will encourage the formation of students spin-offs. For testing the market potential of PhD students topics, I plan to introduce a lecture at PhD level on Technical Entrepreneurship and Innovation. This is already in the curriculum of top universities like UC Berkeley[1].

Regarding **teaching activities**, I will improve the student-centered approach. Definitely, I will continue exposing the students during their early studies to research by proposing summer internships and Bachelor theses with research component. Additionally to the current disciplines taught (Software Engineering, Formal Languages and Automata Theory, and Entrepreneurship Skills - Bachelor level, Formal Verification - Master level), a lecture on Technical Entrepreneurship and Innovation will be proposed at PhD level as mentioned above.

4.2 Upcoming Research in the Near Future

In the near future, numerous scientific endeavors are scheduled to be pursued. These ventures diverge from the current state of personal research findings and ongoing funded projects. The pursuits encompass both theoretical and applied aspects.

Objective 1 Provide an understanding of what the *symmetries* and *similarities* are for the problems which we already studied, i.e resource management in the Cloud and (robustness) verification of binarized neural networks.

Objective 2 Develop methods for breaking the symmetries in and for learning templates from the two problems enumerated above.

Objective 3 Invent theory and algorithms for:

- *abstracting* the symmetries of these case studies using the theory of invariant groups
- *learning problem templates* by formalizing the underlying problem as a graph analysis problem and using graph neural networks (GNNs) for solving it

Objective 4 study the computational effectiveness of the newly developed symmetry breaking and similarity breaking techniques;

Objective 5 for ultimately, performing automatic solving of constraint satisfaction problems (CSP) at scale.

In the following, we will detail the *importance* of fulfilling the objectives above, *difficulties* that we might encounter in the solution process, *limitations of state-of-the-art* and *elements of originality and innovation in relation to the state-of-the-art* of the proposed methodology and methods.

4.3 Importance

In our world of *big data and theoretically intractable problems*, automated preprocessing to simplify formulations of constrained satisfaction problems (CSPs) before or during solving them algorithmically is growing even more important. As previous work shows, suitable preprocessing and simplification techniques [52, 5] as well as

[1] https://startups.berkeley.edu

heuristics [109] have the potential to *reduce computation time from days to seconds*. However, the existing techniques are most of the time experimental and the result of an engineering approach, rather than a methodological one. Hence, a *toolkit of algorithmic processing techniques and heuristics* that reduce the search space and are based on *rigorous mathematical guarantees* is of *utmost importance*. Promising algorithmic preprocessing techniques include *identifying redundant constraints and variables* in the problem formulation, which can be eliminated without changing the answer.

The *two use-cases* that we address are: *(1)* resource provisioning in the Cloud, and *(2)* property verification of binarized neural networks (BNNs). The *main issue* of the existing approaches in solving CSP formalizing *resource provisioning in the Cloud* and *property verification of BNNs* is *scalability*. It is well-known that most of these problems are in NP class, hence solving them is inherently difficult. For example, in the case of a *management system of Cloud resources*, there are various quality and quantity indicators which must be fulfilled, as well as resources of order of hundreds which must be managed. The Cloud Computing community approached these problems using mainly metaheuristics which are known to be approximate methods. This is an issue since the solution is not optimal and one does not know how far from the real solution this is. This could be solved using exact methods. *The benefit of knowing the optimum, would be, for this case study, cost saving.*

Object detection in autonomous driving is performed nowadays using deep neural networks (DNNs). It is well-known that deep learning algorithms are mainly used as a black box, hence they are difficult to debug. In fact, the main criticisms of deep learning algorithms are uncertainty and unexpected behavior on adversarial examples. To overcome this issue, one must perform formal verification of the system DNN, that is, given a DNN and a specification (input and output condition), is there a proof that the DNN satisfies the specification for all inputs? Such *verification guarantees* are urgent and mandatory for *safety-critical systems deploying DNNs*.

4.4 Difficulties

There are mainly two problems when solving a CSP: *(1)* problem formalization, and *(2)* problem solving approach. *Problem formalization* requires deep understanding of the real-world problem as it must capture its relevant parts without introducing useless complexity. It also has significant influence on choosing the problem-solving approach and its computational performance. The issues on the problem-solving approach are related to the *existence* of methods able to solve the problem and to their *scalability*. Typically, CSPs manipulate *polynomial constraints*.

4.5 Limitations of the state-of-the-art

Our research reunites methods from three different areas which manipulate the *same type of objects, polynomials, but using quite different methods. The Satisfiability Checking community* is interested in the problem of determining if a given logical formula is satisfiable, i.e. if there exists an interpretation of certain un-interpreted symbols that evaluates the formula to true. The *SAT Problem* is to check the satisfiability of logical formulae over the Boolean variables and is known to be NP-complete, however powerful SAT solvers have been developed which can solve problems with millions of Boolean variables and are used in many industrial applications (e.g. in program analysis, security, autonomous driving) with socio-economic implications. Motivated by this success, considerable efforts have been made to enrich propositional SAT-solving for different existentially quantified theories (integers, reals, datatypes, etc., and their valid combination) producing *SAT-modulo-theories (SMT) solvers* [16], see [15] for an overview. Although Satisfiability Checking solving has its *strength in efficient techniques for exploring Boolean structures, learning, combining techniques, and developing dedicated heuristics*, it is still *not able to exclusively solve* stringent problems in AI, like resource management [52, 5], or verification of deep neural networks [72, 87]. Notable is the fact that many of the AI problems, *(1) exhibit symmetries* and/or *(2) are similar*. Hence, the idea is to exploit *symmetries* by breaking them and/or *similarities* by learning problem templates. Eliminating (some of) the symmetries, task known as *symmetry breaking*, would reduce the search space of solutions, hence will make the problem easier to solve. It has been used to solve many NP-hard problems [52, 5], however, *no systematic study on the techniques applied exists in the context of real-world problems, all the advances being experimental*, by trial and error[2]. In practice, it is common that *problems from the same family*, hence *similar*, are solved. For example, for *resource provisioning in the Cloud*, typically only some of components' hardware requirements change or there is an increasing/decreasing number of instances. In this case, one can benefit from *learning problem templates* to speed-up the computational time. These will be approached by using methods from *Symbolic Computation (for symmetry breaking) and Machine Learning (for similar problems)*. The *Symbolic Computation (or Computer Algebra)* community has been using computers to *solve hard non-linear problems* basically at the same time, but independently to the satisfiability checking community. It includes solving non-linear problem over both the real and complex numbers, though generally with very different techniques. This has produced many *new applications and surprising developments*, for example methods of Groebner Basis (GB) [26] and Cylindrical Algebraic Decomposition (CAD) [37] have been used for the development of the Todai robot [9]. Although Symbolic Computation provides powerful procedures for sets of arithmetic constraints and has expertise in *simplification* and *preprocessing*, the algorithms developed address large classes of problems, thus *have high complexity.* David Hilbert developed *invariant theory* while he was working on symmetry breaking problems from physics. Theory of *invariants of finite groups* has developed techniques for describing all polynomials that are unchanged when we change variables according to a given finite group of matrices, thus it can be used to reason about symmetries. This theory has not been exploited from the point of view of computational benefits it might have in solving real world problems. The Satisfiability Checking and Symbolic Computation communities *have joint forces* to solve problems that were impossible to be solved by the two communities separately. Such example is the SC-square: Satisfiability Checking and Symbolic Computation[15] project whose importance was acknowledged by receiving

[2] For the cloud resource management problem, a bin-packing problem, methods for symmetry breaking follow, e.g. [93]

funding from the European Union's Horizon 2020 research and innovation program. The idea of *combining the two different research disciplines* for solving challenging problems is not new. For example,

- algebraic techniques (e.g. GB, CAD, Gosper algorithm, C-finite solving) have been used to generate loop invariants [76, 95, 97],
- *we have previously used computational logic and quantifier elimination by cylindrical algebraic decomposition to synthesize optimal numerical schemes* [47, 48],
- machine learning has been used to improve CAD [66].

These are just some examples. Moreover, prestigious ERC Starting Grant[3] has been awarded in 2014 to efforts on bringing symbolic computation and automated reasoning together to solve an important problem of safety-critical systems, namely correctness analysis.

Besides the combination of the above-mentioned approaches which will be mainly used, we aim to apply GNNs[4] [62, 98], to speed-up the solution process for problems which are in the same family, hence similar. This approach is suitable only for the problem of Cloud resource management as it exhibits a relational structure like the graphs. To the best of our knowledge GNNs have not been exploited for this problem, but have successfully been used to speed-up the solution process for other NP hard problems [29].

Summarizing, the *limitations* of the state-of-the-art are as follows.

- The techniques of Satisfiability Checking are not able to efficiently solve problems exhibiting symmetries or do not consider if problems belong to the same family, although the last decade progress towards computationally effective tools.
- Symbolic Computation developed the theory of invariants but showed its effectiveness mainly on toy examples [42].
- The theory of invariants has not been exploited for advancing the computational capabilities of the satisfiability checkers, even research combining satisfiability and symbolic computation exists.
- GNNs were insufficiently exploited to solve problems from Cloud resource management domain and there has not been any attempt to combine them with SAT/SMT solving.

4.6 Elements of originality and innovation in relation to the state-of-the-art

Combining Satisfiability Checking and Symbolic Computation has an old research tradition, however *our idea of combining SAT/SMT solving and invariant theory*

[3] Laura Kovacs, Symbolic Computation and Automated Reasoning for Program Analysis: https://cordis.europa.eu/project/rcn/197841/factsheet/de

[4] GNNs are a hot research topic (see `https://tinyurl.com/GNNsICLR2021`)

for symmetry breaking is novel. We believe that the newly developed theory and algorithms will be able to solve timely problems, as those proposed by us. We will perform a *systematic study* of how invariant theory could be combined with state-of-the-art satisfiability checkers which *will be validated on two real-world problems.* On the other hand, to the best of our knowledge, *the idea of combining GNNs with SAT/SMT solving for Cloud resource management is new* and corresponds to the AIOps phase of a system to be deployed in Cloud. In fact, the SpecOps phase was already addressed in our projects MANeUveR and SAGE and proposee to be refined further (see Figure 4.1).

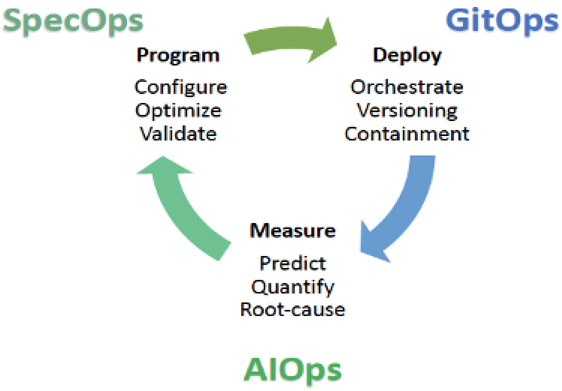


Figure 4.1: Nikolaj Bjørner. Infusing Azure with Verifiable Reliability. Invited talk FoFoSDN 2020

We propose the following methodology for solving GNNs-based problems.

1. The formulation of the problem as a classical graph structure (nodes are application components while edges specify if there exists a relation between two components, for example, if two components need to be placed on the same VM). Additionally, nodes are augmented with features (for example, how much hardware requirements are necessary for the respective component, in which kind of relations the component is with the others).

2. Based on the node features and the graph structure, we aim to learn a representation of the graph at, for example, graph level. This will facilitate deriving/predicting properties at graph level (e.g. optimal price and assignment of components to the VMs). Next, SAT/SMT solvers will be used to recalibrate the property value as the learning process is an approximating one.

Bibliography

[1] Belgian Traffic Sign Database. `https://www.kaggle.com/datasets/shazaelmorsh/trafficsigns`. Accessed: March 25th, 2023.

[2] Chinese Traffic Sign Database. `https://www.kaggle.com/datasets/dmitryyemelyanov/chinese-traffic-signs`. Accessed: March 25th, 2023.

[3] German Traffic Sign Recognition Benchmark. `https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign?datasetId=82373&language=Python`. Accessed: March 25th, 2023.

[4] The Zot bounded satisfiability checker. github.com/fm-polimi/zot.

[5] Erika Ábrahám, Florian Corzilius, Einar Broch Johnsen, Gereon Kremer, and Jacopo Mauro. Zephyrus2: on the fly deployment optimization using SMT and CP technologies. In *Dependable Software Engineering: Theories, Tools, and Applications - Second International Symposium, SETTA 2016, Beijing, China, November 9-11, 2016, Proceedings*, pages 229–245, 2016.

[6] Gotz Alefeld and Jurgen Herzberger. *Introduction to interval computations*. Academic Press, Inc., New York, NY, 1983.

[7] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[8] Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, and Ming Xu. Web services agreement specification (ws-agreement). In *Open grid forum*, volume 128, page 216. Citeseer, 2007.

[9] Noriko H Arai. The impact of AI — can a robot get into the university of tokyo? *National Science Review*, 2(2):135–136, 2015.

[10] Bashar Al Asaad and Madalina Erascu. A tool for fake news detection. In *20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2018, Timisoara, Romania, September 20-23, 2018*, pages 379–386, 2018.

[11] Henna Asikainen. Successful learning and studying in biosciences. *Exploring how students' conceptions of learning, approaches to learning, motivation and their experiences of the teaching-learning environment are related to study success. Helsinki: Unigrafia*, 2014.

[12] Henna Asikainen and David Gijbels. Do students develop towards more deep approaches to learning during studies? a systematic review on the development of students' deep and surface approaches to learning in higher education. *Educational Psychology Review*, 29(2):205–234, 2017.

[13] Marlies Baeten, Eva Kyndt, Katrien Struyven, and Filip Dochy. Using student-centred learning environments to stimulate deep approaches to learning: Factors encouraging or discouraging their effectiveness. *Educational research review*, 5(3):243–260, 2010.

[14] Timothy T Baldwin and J Kevin Ford. Transfer of training: A review and directions for future research. *Personnel psychology*, 41(1):63–105, 1988.

[15] Clark Barrett, Daniel Kroening, and Thomas Melham. Problem solving for the 21st century: Efficient solver for satisfiability modulo theories. 2014.

[16] Clark Barrett and Cesare Tinelli. *Satisfiability modulo theories*. Springer, 2018.

[17] N. Beebe. Accurate square root computation. Technical report, Center for Scientific Computing, Department of Mathematics, University of Utah, 1991.

[18] Karin Bernsmed, Martin Gilje Jaatun, Per Hakon Meland, and Astrid Undheim. Security slas for federated cloud services. In *2011 Sixth International Conference on Availability, Reliability and Security*, pages 202–209. IEEE, 2011.

[19] Marcello M Bersani, Francesco Marconi, Matteo Rossi, and Madalina Erascu. A tool for verification of big-data applications. In *Proceedings of the 2nd International Workshop on Quality-Aware DevOps, QUDOS@ISSTA 2016, Saarbrücken, Germany, July 21, 2016*, pages 44–45, 2016.

[20] Marcello M Bersani, Francesco Marconi, Matteo Rossi, Madalina Erascu, and Silvio Ghilardi. Formal verification of data-intensive applications through model checking modulo theories. In *Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software*, SPIN 2017, pages 98–101, New York, NY, USA, 2017. ACM.

[21] Marcello M. Bersani, Matteo Rossi, and Pierluigi San Pietro. A tool for deciding the satisfiability of continuous-time metric temporal logic. In *Proc. of TIME*, pages 99–106, 2013.

[22] Marcello M Bersani, Matteo Rossi, and Pierluigi San Pietro. A tool for deciding the satisfiability of continuous-time metric temporal logic. *Acta Informatica*, 53(2):171–206, 2016.

[23] Marcello M. Bersani, Matteo Rossi, and Pierluigi San Pietro. A tool for deciding the satisfiability of continuous-time metric temporal logic. *Acta Informatica*, 53(2):171–206, 2016.

[24] John Biggs, David Kember, and Doris YP Leung. The revised two-factor study process questionnaire: R-spq-2f. *British journal of educational psychology*, 71(1):133–149, 2001.

[25] Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. νZ - an optimizing SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, London, UK, April 11-18, 2015. Proceedings*, pages 194–199, 2015.

[26] NK Bose. *Gröbner bases: An algorithmic method in polynomial ideal theory.* Springer, 1995.

[27] Ahmed Bouajjani and Richard Mayr. Model checking lossy vector addition systems. In *Proceedings of STACS*, volume 1563 of *LNCS*, pages 323–333, 1999.

[28] Christopher W Brown. QEPCAD-B: A program for computing with semi-algebraic sets using cads. *SIGSAM Bulletin*, 37(4):97–108, 2003.

[29] Quentin Cappart, Didier Chételat, Elias B Khalil, Andrea Lodi, Christopher Morris, and Petar Velickovic. Combinatorial optimization and reasoning with graph neural networks. *J. Mach. Learn. Res.*, 24:130–1, 2023.

[30] Giuliano Casale, Danilo Ardagna, Matej Artac, Franck Barbier, Elisabetta Di Nitto, Alexis Henry, Gabriel Iuhasz, Christophe Joubert, Jose Merseguer, Victor Ion Munteanu, Juan Perez, Dana Petcu, Matteo Rossi, Chris Sheridan, Ilias Spais, and Daniel Vladušič. DICE: Quality-driven development of data-intensive cloud applications. In *Proc. of MiSE*, pages 78–83, 2015.

[31] Valentina Casola, Alessandra De Benedictis, Madalina Erascu, Jolanda Modic, and Massimiliano Rak. Automatically enforcing security SLAs in the cloud. *IEEE Trans. Services Computing*, 10(5):741–755, 2017.

[32] Valentina Casola, Alessandra De Benedictis, Madalina Erascu, Massimiliano Rak, and Umberto Villano. A security SLA-driven methodology to set-up security capabilities on top of Cloud services. In *10th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2016, Fukuoka, Japan, July 6-8, 2016*, pages 549–554, 2016.

[33] Valentina Casola, Alessandra De Benedictis, and Massimiliano Rak. On the adoption of security slas in the cloud. *Accountability and Security in the Cloud: First Summer School, Cloud Accountability Project, A4Cloud, Malaga, Spain, June 2-6, 2014, Revised Selected Papers and Lectures 1*, pages 45–62, 2015.

[34] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 9–16. IEEE, 2016.

[35] Geoffrey Chu, Peter J. Stuckey, Andreas Schutt, Thorsten Ehlers, and Kathryn Francis Graeme Gange. Chuffed, a lazy clause generation solver, 2022. Last accessed 20 August 2022.

[36] William James Cody. *Software Manual for the Elementary Functions (Prentice-Hall series in computational mathematics).* Prentice-Hall, Inc., 1980.

[37] George E Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition: a synopsis. *ACM SIGSAM Bulletin*, 10(1):10–12, 1976.

[38] Bogdan David and Madalina Erascu. Benchmarking optimization solvers and symmetry breakers for the automated deployment of component-based applications in the cloud (extended abstract), 2023. Presented at 7th International Workshop on Satisfiability Checking and Symbolic Computation (SC-square), Part of IJCAR 22, at FLOC 2022, August 12, 2022, Haifa, Israel.

[39] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[40] Catherine De Rijdt, Ann Stes, Cees Van Der Vleuten, and Filip Dochy. Influencing variables and moderators of transfer of learning to the workplace within the area of staff development in higher education: Research review. *Educational Research Review*, 8:48–74, 2013.

[41] M Dekker and G Hogben. Survey and analysis of security parameters in cloud slas across the european public sector. *Survey and analysis of security parameters in cloud SLAs across the European public sector*, 2011.

[42] Harm Derksen and Gregor Kemper. *Computational invariant theory*. Springer, 2015.

[43] Andreas Dolzmann and Thomas Sturm. Redlog: Computer algebra meets computer logic. *Acm Sigsam Bulletin*, 31(2):2–9, 1997.

[44] Mădălina Eraşcu. *Computational Logic and Quantifier Elimination Techniques for (Semi-)automatic Static Analysis and Synthesis of Algorithms*. PhD thesis, Research Institute for Symbolic Computation, 2012. RISC Technical Report 12-16.

[45] Madalina Erascu. Efficient simplification techniques for special real quantifier elimination with applications to the synthesis of optimal numerical algorithms. In *International Workshop on Computer Algebra in Scientific Computing*, pages 193–211. Springer, 2016.

[46] Madalina Erascu and Hoon Hong. The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation. *Journal of Reliable Computing*, 18:73–81, 2013.

[47] Madalina Erascu and Hoon Hong. Synthesis of optimal numerical algorithms using real quantifier elimination (case study: Square root computation). In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 162–169, 2014.

[48] Madalina Erascu and Hoon Hong. Real quantifier elimination for the synthesis of optimal numerical algorithms (case study: Square root computation). *Journal of Symbolic Computation*, 75:110–126, 2016. Special issue on the conference ISSAC 2014: Symbolic computation and computer algebra.

[49] Madalina Erascu, Gabriel Iuhasz, and Flavia Micota. An architecture for a management agency for cloud resources. In *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 288–295. IEEE, 2018.

[50] Madalina Erascu and Razvan Metes. Constrained optimization benchmark for optimization modulo theory: A cloud resource management problem. In *SMT 2019 17th International Workshop on Satisfiability Modulo Theories, affiliated with SAT 2019, July 7-8, 2019, Lisbon, Portugal*, 2019.

[51] Madalina Erascu, Flavia Micota, and Daniela Zaharie. Influence of variables encoding and symmetry breaking on the performance of optimization modulo theories tools applied to cloud resource selection. In Gilles Barthe, Konstantin Korovin, Stephan Schulz, Martin Suda, Geoff Sutcliffe, and Margus Veanes, editors, *LPAR-22 Workshop and Short Paper Proceedings*, volume 9 of *Kalpa Publications in Computing*, pages 1–14. EasyChair, 2018.

[52] Madalina Erascu, Flavia Micota, and Daniela Zaharie. Scalable optimal deployment in the cloud of component-based applications using optimization modulo theory, mathematical programming and symmetry breaking. *J. Log. Algebraic Methods Program.*, 121:100664, 2021.

[53] Madalina Erascu and V. Mladenovici. Transferring learning into the workplace: Evaluating a student-centered learning approach through computer science students' lens. In Mutlu Cukurova, Nikol Rummel, Denis Gillet, Bruce M. McLaren, and James Uhomoibhi, editors, *Proceedings of the 14th International Conference on Computer Supported Education, CSEDU 2022, Online Streaming, April 22-24, 2022, Volume 2*, pages 442–449. SCITEPRESS, 2022.

[54] David Fowler and Eleanor Robson. Square root approximations in old babylonian mathematics: Ybc 7289 in context. *Historia Mathematica*, 25(4):366–378, 1998.

[55] Michael Fullan. Staff development, innovation, and institutional development. *Changing school culture through staff development*, 1220:16–133, 1990.

[56] Gecode Team. Gecode: Generic constraint development environment, 2006. Available from `http://www.gecode.org`.

[57] Lukas Geiger and Plumerai Team. Larq: An open-source library for training binarized neural networks. *Journal of Open Source Software*, 5(45):1746, 2020.

[58] Gerry Geitz, Desirée Joostenten Brinke, and Paul A Kirschner. Changing learning behaviour: Self-efficacy and goal orientation in PBL groups in higher education. *International Journal of Educational Research*, 75:146–158, 2016.

[59] Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, and Daniele Zucchelli. Towards SMT model checking of array-based systems. In *Proc. of IJCAR*, pages 67–82, 2008.

[60] Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, and Daniele Zucchelli. Towards smt model checking of array-based systems. In *Automated Reasoning: 4th International Joint Conference, IJCAR 2008 Sydney, Australia, August 12-15, 2008 Proceedings 4*, pages 67–82. Springer, 2008.

[61] Graham Gibbs and Martin Coffey. The impact of training of university teachers on their teaching skills, their approach to teaching and the approach to learning of their students. *Active learning in higher education*, 5(1):87–100, 2004.

[62] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[63] Paul Ginns, Michael Prosser, and Simon Barrie. Students' perceptions of teaching quality in higher education: The perspective of currently enrolled students. *Studies in higher education*, 32(5):603–615, 2007.

[64] Joint Task Force Transformation Initiative Interagency Working Group et al. Nist special publication 800-53 revision 4-security and privacy controls for federal information systems and organizations. *National Institute of Standards and Technology, Technical rep*, 2013.

[65] John F Hart. *Computer approximations*. Krieger Publishing Co., Inc., 1978.

[66] Zongyan Huang, Matthew England, David J Wilson, James Bridge, James H Davenport, and Lawrence C Paulson. Using machine learning to improve cylindrical algebraic decomposition. *Mathematics in Computer Science*, 13:461–488, 2019.

[67] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016.

[68] Marian D Ilie, Nicolae Aurelian Bibu, Adriana Isac, Velibor Mladenovici, and Daniel Emil Iancu. Raport intermediar 1 (Ri1) privind evaluarea pretest a cadrelor didactice și a studenților din grupul țintă [Intermediate Report 1 (Ri1) on the pretest assessment of academics and students in the target group], 2021. https://cda.uvt.ro/documents/.

[69] Marian D Ilie, Laurențiu P Maricuțoiu, Daniel E Iancu, Izabella G Smarandache, Velibor Mladenovici, Dalia CM Stoia, and Silvia A Toth. Reviewing the research on instructional development programs for academics. trying to tell a different story: A meta-analysis. *Educational Research Review*, 30:100331, 2020.

[70] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[71] Andrei Iovescu. Benchmark problems for the constraints satisfaction problems repository. *STUDMath-IT 2020*, 2020.

[72] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017.

[73] David Kember. Promoting student-centred forms of learning across an entire university. *Higher education*, 58(1):1–13, 2009.

[74] David Kember and Doris YP Leung. Establishing the validity and reliability of course evaluation questionnaires. *Assessment & Evaluation in Higher Education*, 33(4):341–353, 2008.

[75] Donald Kirkpatrick and James Kirkpatrick. *Evaluating training programs: The four levels*. Berrett-Koehler Publishers, 2006.

[76] Laura Kovács. Reasoning algebraically about p-solvable loops. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 249–264. Springer, 2008.

[77] Petra Lietz and Bobbie Matthews. The effects of college students' personal values on changes in learning approaches. *Research in higher education*, 51(1):65–87, 2010.

[78] Francesco Marconi, Marcello M. Bersani, Madalina Erascu, and Matteo Rossi. Towards the formal verification of data-intensive applications through metric temporal logic. In Kazuhiro Ogata, Mark Lawford, and Shaoying Liu, editors, *Formal Methods and Software Engineering*, pages 193–209, Cham, 2016. Springer International Publishing.

[79] Herbert W Marsh. Do university teachers become more effective with experience? A multilevel growth model of students' evaluations of teaching over 13 years. *Journal of educational psychology*, 99(4):775, 2007.

[80] John E Meggitt. Pseudo division and pseudo multiplication processes. *IBM Journal of Research and Development*, 6(2):210–226, 1962.

[81] Deborah S Meizlish, Mary C Wright, Joseph Howard, and Matthew L Kaplan. Measuring the impact of a new faculty program using institutional data. *International Journal for Academic Development*, 23(2):72–85, 2018.

[82] Flavia Micota, Madalina Erascu, and Daniela Zaharie. Constraint satisfaction approaches in Cloud resource selection for component based applications. In *14th IEEE International Conference on Intelligent Computer Communication and Processing, ICCP 2018, Cluj-Napoca, Romania, September 6-8, 2018*, pages 443–450, 2018.

[83] Microsoft Research. Z3: An efficient SMT solver. z3.codeplex.com.

[84] Ramon E Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966.

[85] Ramon E Moore, R Baker Kearfott, and Michael J Cloud. *Introduction to interval analysis*. SIAM, 2009.

[86] DR Morrison. A method for computing certain inverse functions. *Mathematical Tables and Other Aids to Computation*, 10(56):202–208, 1956.

[87] Nina Narodytska. Formal Analysis of Deep Binarized Neural Networks. In *IJCAI*, pages 5692–5696, 2018.

[88] Nicholas Nethercote, Peter J. Stuckey, Ralph Becket, Sebastian Brand, Gregory J. Duck, and Guido Tack. Minizinc: Towards a standard CP modelling language. In Christian Bessiere, editor, *Principles and Practice of Constraint Programming - CP 2007, 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007, Proceedings*, volume 4741 of *Lecture Notes in Computer Science*, pages 529–543. Springer, 2007.

[89] Laurent Perron and Vincent Furnon. Or-tools. *Google.[Online]. Available: https://developers. google. com/optimization*, 2019.

[90] Liisa Postareff, Sari Lindblom-Ylänne, and Anne Nevgi. A follow-up study of the effect of pedagogical training on teaching in higher education. *Higher Education*, 56(1):29–43, 2008.

[91] Andreea Postovan and Madalina Erascu. Architecturing binarized neural networks for traffic sign recognition, 2023. To appear in Proceedings of 32nd International Conference on Artificial Neural Networks.

[92] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: ImageNet Classification using Binary Convolutional Neural Networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[93] Jean-Charles Régin and Mohamed Rezgui. Discussion about constraint programming bin packing models. *AI for data center management and cloud computing*, 11:08, 2011.

[94] Nathalie Revol. Interval Newton iteration in multiple precision for the univariate case. *Numerical Algorithms*, 34(2–4):417–426, 2003.

[95] Enric Rodríguez-Carbonell and Deepak Kapur. Automatic generation of polynomial loop invariants: Algebraic foundations. In *Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 266–273, 2004.

[96] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[97] Sriram Sankaranarayanan, Henny B Sipma, and Zohar Manna. Non-linear loop invariant generation using gröbner bases. In *Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 318–329, 2004.

[98] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[99] Martina Seidl, Marion Scholz, Christian Huemer, and Gerti Kappel. *UML@ classroom: An introduction to object-oriented modeling.* Springer, 2015.

[100] Izabella G Smarandache, Laurentiu P Maricutoiu, Marian D Ilie, Daniel E Iancu, and Velibor Mladenovici. Students' approach to learning: evidence regarding the importance of the interest-to-effort ratio. *Higher Education Research & Development*, pages 1–16, 2021.

[101] Ann Stes, Liesje Coertjens, and Peter Van Petegem. Instructional development for teachers in higher education: Impact on teaching approach. *Higher education*, 60(2):187–204, 2010.

[102] Ann Stes, Liesje Coertjens, and Peter Van Petegem. Instructional development in higher education: Impact on teachers' teaching behaviour as perceived by students. *Instructional Science*, 41(6):1103–1126, 2013.

[103] Ann Stes, Sven De Maeyer, David Gijbels, and Peter Van Petegem. Instructional development for teachers in higher education: Effects on students' perceptions of the teaching–learning environment. *British Journal of Educational Psychology*, 82(3):398–419, 2012.

[104] Ann Stes, Mariska Min-Leliveld, David Gijbels, and Peter Van Petegem. The impact of instructional development in higher education: The state-of-the-art of the research. *Educational research review*, 5(1):25–49, 2010.

[105] IBM Team. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual. Version 12*, 2016.

[106] Hans Van Vliet, Hans Van Vliet, and JC Van Vliet. *Software engineering: principles and practice*, volume 13. John Wiley & Sons Hoboken, NJ, 2008.

[107] J. H. Wensley. A class of non-analytical iterative processes. *The Computer Journal*, 1(4):163–167, 1959.

[108] Inc. Wolfram Research. *Mathematica Edition: Version 8.0.* Wolfram Research, Inc., 2010.

[109] Jiangtao Zhang, Hejiao Huang, and Xuan Wang. Resource provision algorithms in cloud computing: A survey. *Journal of network and computer applications*, 64:23–42, 2016.