Volume 26 Number 2

ACTA CYBERNETICA

Szeged, 2023

ACTA CYBERNETICA

Information for authors. Acta Cybernetica publishes only original papers in the field of Computer Science. Manuscripts must be written in good English. Contributions are accepted for review with the understanding that the same work has not been published elsewhere. Papers previously published in conference proceedings, digests, preprints are eligible for consideration provided that the author informs the Editor at the time of submission and that the papers have undergone substantial revision. If authors have used their own previously published material as a basis for a new submission, they are required to cite the previous work(s) and very clearly indicate how the new submission offers substantively novel or different contributions beyond those of the previously published work(s). There are no page charges. An electronic version of the published paper is provided for the authors in PDF format.

Manuscript Formatting Requirements. All submissions must include a title page with the following elements: title of the paper; author name(s) and affiliation; name, address and email of the corresponding author; an abstract clearly stating the nature and significance of the paper. Abstracts must not include mathematical expressions or bibliographic references.

References should appear in a separate bibliography at the end of the paper, with items in alphabetical order referred to by numerals in square brackets. Please prepare your submission as one single PostScript or PDF file including all elements of the manuscript (title page, main text, illustrations, bibliography, etc.).

When your paper is accepted for publication, you will be asked to upload the complete electronic version of your manuscript. For technical reasons we can only accept files in LaTeX format. It is advisable to prepare the manuscript following the guidelines described in the author kit available at `https://cyber.bibl.u-szeged.hu/index.php/actcybern/about/submissions` even at an early stage.

Submission and Review. Manuscripts must be submitted online using the editorial management system at `https://cyber.bibl.u-szeged.hu/index.php/actcybern/submission/wizard`. Each submission is peer-reviewed by at least two referees. The length of the review process depends on many factors such as the availability of an Editor and the time it takes to locate qualified reviewers. Usually, a review process takes 6 months to be completed.

Subscription Information. Acta Cybernetica is published by the Institute of Informatics, University of Szeged, Hungary. Each volume consists of four issues, two issues are published in a calendar year. Subscription rates for one issue are as follows: 5000 Ft within Hungary, €40 outside Hungary. Special rates for distributors and bulk orders are available upon request from the publisher. Printed issues are delivered by surface mail in Europe, and by air mail to overseas countries. Claims for missing issues are accepted within six months from the publication date. Please address all requests to:

Acta Cybernetica, Institute of Informatics, University of Szeged
P.O. Box 652, H-6701 Szeged, Hungary
Tel: +36 62 546 396, Fax: +36 62 546 397, Email: `acta@inf.u-szeged.hu`

Web access. The above information along with the contents of past and current issues are available at the Acta Cybernetica homepage `https://cyber.bibl.u-szeged.hu/` .

Tibor Gyimóthy
Department of Software Engineering, University of Szeged, Hungary
gyimothy@inf.u-szeged.hu

Zoltan Kato
Department of Image Processing and Computer Graphics, University of Szeged, Hungary
kato@inf.u-szeged.hu

Dragan Kukolj
RT-RK Institute of Computer Based Systems, Novi Sad, Serbia
dragan.kukolj@rt-rk.com

László Lovász
Department of Computer Science, Eötvös Loránd University, Budapest, Hungary
lovasz@cs.elte.hu

Kálmán Palágyi
Department of Image Processing and Computer Graphics, University of Szeged, Hungary
palagyi@inf.u-szeged.hu

Dana Petcu
Department of Computer Science, West University of Timisoara, Romania
petcu@info.uvt.ro

Andreas Rauh
School II – Department of Computing Science, Group Distributed Control in Interconnected Systems, Carl von Ossietzky Universität Oldenburg, Germany
andreas.rauh@uni-oldenburg.de

Heiko Vogler
Department of Computer Science, Dresden University of Technology, Germany
Heiko.Vogler@tu-dresden.de

Acta Cybernetica **26** (2023) 149–173.

EA-POT: An Explainable AI Assisted Blockchain Framework for Honeypot IP Predictions*

Shajulin Benedict[a]

Abstract

The culpable cybersecurity practices that threaten leading organizations are logically prone to establishing countermeasures, including honeypots, and bestowing research innovations in various dimensions, such as ML-enabled threat predictions. This article proposes an explainable AI-assisted permissioned blockchain framework named `EA-POT` for predicting potential defaulters' IP addresses. `EA-POT` registers the probable defaulters predicted by explainable AI based on the approval of IP authorizers of blockchain databases. Experiments were carried out at the IoT Cloud Research laboratory using three prediction models, such as Random Forest Modeling (RFM), Linear Regression Modeling (LRM), and Support Vector Machines (SVM); and, the experimental results for predicting the AWS honeypots were explored. The proposed `EA-POT` framework revealed the procedure for including interpretable knowledge while blacklisting IPs that reach honeypots.

Keywords: blockchain, cyber security, honeypot, hyperledger, predictions

1 Introduction

Developing a secure cloud-based or IoT-enabled application is an extraordinary feat of development as newer security issues evolve, especially when the post COVID-19 scenario was considered in a connected devices world. Remote accesses to organizational resources and services are prone to security challenges in newer dimensions. Notably, as an essential part of preparedness, transferring identity credentials to employees has become a landmark shift in handling the security challenge needed to protect resources.

It is estimated by high-income companies/organizations and researchers that a reasonably high volume of budget needs to be spent to counteract evolving cybersecurity issues. For instance, Australia economists have estimated that it will spend over $7.6 billion by 2024 [32]; Investments towards cloud security tools are projected

*This work was supported by AIC IIITKottayam and BEL Consultancy Project.

[a]Indian Institute of Information Technology Kottayam, India, E-mail: `shajulin@iiitkottayam.ac.in`, ORCID: `0000-0002-2543-2710`, www.sbenedictglobal.com

DOI: `10.14232/actacyb.293319`

to increase from $5.6 billion in 2018 to $12.6 billion in 2024 [36]; `Centrify`, a company specializing in cybersecurity, highlighted the possibility of phishing attacks that could lead to a huge investment in potential IIoT industries [35].

Looking to the future, in the wake of COVID-19, many countries or organizations, especially those belonging to government sectors, have suggested newer security policies or procedures to counteract notable sprouting security challenges, such as i) phishing, ii) malicious attacks, iii) accessing orphaned accounts, iv) ransomware attacks, v) advanced persistent threat, and so forth. In fact, the *WannaCry* ransomware attack challenged over 150 countries [17]. Additionally, IoT devices, which increase day by day, require diligent secure connectivity services. The failure to provide proactive secured access to connected devices or associated cloud services, especially in the automobile and healthcare industries, could lead to unnecessary data leaks. This could disrupt important automated decisions and slow down the global economic situation. There have been a few research efforts in the recent past to address the IoT security inefficiencies [26, 27, 31].

Obviously, preventing potential attackers/hackers from breaching security needs to be handled diligently. In recent years, honeypots have been established by several leading cloud-based service providers, including AWS IoT infrastructure providers. Honeypot, in general, lies alongside the firewall inviting security challenges from potential hackers. In doing so, the specific pattern of attacks can be explored; the motivation of cybercriminals in writing code could be reduced; the activity of investing money for illegal purposes may be minimized; the intention of attackers and involved countries can be observed; and, the possibility of the attackers' evolving innovations can also be studied.

Traditionally, honeypots on cloud infrastructures address several known issues as listed below:

1. The attacks initiated by their own organizations' employees must be diligently handled. In fact, such organizational attacks are possible due to poor knowledge of utilizing cyber-physical devices/gadgets or the associated services;

2. Indigent policies of honeypots need to be dynamically handled in a decentralized environment ; and,

3. The time needed to learn about the potential attack has to be negligible compared to the time it takes to attack.

This paper proposes an `EA-POT` framework, an Explainable AI-assisted blockchain framework, for honeypot IP predictions. `EA-POT` attempts to reduce the time needed to identify potential attackers using prediction algorithms, such as Random Forest Modeling (RFM), Linear Regression Modeling (LRM), and Support Vector Machines (SVM). Unlike traditional methods, which are dependent on non-explainable parameters (black-box and temperamental), the proposed `EA-POT` framework enables the explainability features of prediction models.

The framework is combined with a hyperledger fabric-based blockchain network to register the honeypot IP addresses and to inject dynamic prevention policies on

the fly. The reasons for including the permissioned blockchain into the framework are multi-fold:

1. The blacklisted IP addresses considered to be more vulnerable become immutable as the organizations or the inner employees of an organization cannot modify them; and,
2. Specific policies can be formulated by involving permissioned organizations or stakeholders in deciding the actions against the defaulters.

In addition, experts believe that the performance of the hyperledger fabric, especially when the chaincodes are written using golang, is reasonably better than the other blockchains. Authors of [9] have studied the performance impact of transactions concerning the underlying programming languages; similarly, authors of [22] have delved into the end-to-end transaction latency factors of hyperledger fabric blockchains.

In this paper, the research work emphasized the importance of the `EA-POT` framework to register blacklisted IPs, which were explainable using prediction models, in the immutable database. Experiments were held at the IoT cloud research laboratory on distributed systems after a Kubernetes cluster of hyperledger fabric components was launched.

The major contributions of the work are listed as follows:

1. an `EA-POT` framework was proposed to register potential hackers into the blockchain database after the policies were satisfied;
2. the importance of explainable AI while predicting IPs was explored; and,
3. the experimental results were investigated and revealed to highlight the necessity of the proposed `EA-POT` framework.

The rest of the paper is organized as follows: Section 2 investigates the state-of-the-art research in the field of honeypots and the utilization of explainable AI for enhancing cybersecurity; Section 3 reveals the functionalities and components of the proposed `EA-POT` framework; Section 4 illustrates the approach of utilizing explainable AI for the framework; Section 6 manifests the experimental evaluations of the proposed framework that were carried out at the laboratory; and finally, Section 7 offers a few outlooks and conclusions for the near future research based on the proposed work.

2 Related Work

Countering cybercrime in several countries is often considered an ongoing crucial agenda. In fact, a proactive approach to handling security measures has attracted several researchers/countries in recent years. Honeypots, being a measure of luring potential hackers, have served as a foundation for proactively analyzing the characteristics of hackers and their malicious behaviors.

This section explains the state-of-the-art work of honeypot research in three different perspectives as listed below:

1. Honeypot placements (Clouds),
2. Inclusion of Machine Learning / Explainable AI,
3. Application of Blockchains.

Finally, the shortcomings of the existing works and the contributions of this article are expressed in the section.

2.1 Honeypot Research – Domains, Placements, and Clouds

Researchers/practitioners belonging to several domains, such as Clouds and Industrial IoT (IIoT), have evidenced the importance of including honeypots in their organizations. There exist several honey pot implementations, both static and dynamic, in IIoT or cloud environments. For instance, authors of [20] and [16] studied the application of honeypots for smart grids; authors of [7] revealed the importance of honeypots for capturing DDoS attacks in IIoT environments; and, authors of [24] proposed a social leopard algorithm to detect ransomware attacks using honeypots. Additionally, a few honey pot implementations for protecting buildings [3] and establishing a secured smart home infrastructure [14] have been developed in the recent past.

A sector of researchers has attempted to optimize honeypot placements in organizations based on malicious attackers – i.e., authors of [12] and [1] have applied a game-theoretic framework model to optimally choose honeypots in various locations; in [10], the authors have installed honeypots in nine countries and studied the behavior of malicious users. Besides, honeypots have been widely deployed to enable lightweight interactions in IoT-based infrastructures. For instance, authors of [26] have implemented *BoTNet*, and authors of [27] have implemented *IoTCMal* for low interaction honeypots using TelNet and SSH; authors of [6] have deployed a global honeypot infrastructure to detect industrial attacks.

The deployment of honeypots has been studied in cloud environments as potential attacks on public cloud infrastructures, such as AWS Cloud, Google Compute Engine, and Microsoft Azure. This process has become an inevitable activity. Accordingly, a few researchers have oriented their analysis and studies towards cloud infrastructures. For instance, the authors of [21] have developed a high interaction system using Kerberos authentication, Virtual Private Cloud, and Elastic File System to understand the malicious nature of attackers; the authors of [13] have developed honeypot as a service model for luring attackers. This honeypot-as-a-service is implemented as a plug-and-play model, which could be hosted on gateways for capturing the malicious attackers; and, a few practitioners have listed the AWS honeypot data that suggested potential hackers, who attempted to maliciously attack AWS cloud services, including AWS IoT services.

2.2 Machine Learning and Explainable AI

Traditionally, machine learning has been applied in several domains, including IoT to predict machine behavior or future events [8]. Several variants of machine learning, including federated learning aspects, have reached the market for efficient learning processes of IoT and cloud services [28, 4]. Additionally, the application of own decision-making algorithms, such as neural networks, has been practiced to classify security attacks [2].

In fact, proactively learning the behavior of malicious attackers or potential IP addresses of the attacker's needs a diligent skillset that modern computational efforts are required. With the evolution of several machine learning platforms and tools, in recent years, the identification of attackers and the classification of the severity of attacks have become a widely discussed topic of research. Authors of [23] have studied the application of probabilistic models for proactively estimating the honeypot detection. The authors have confined their research to TELNET and SSH-based communications in IoT domains. Authors of [15] have applied an outlier detection mechanism to project anomalies from the outlier information. To do so, the authors have utilized unsupervised machine learning approaches for honeypots.

A few machine learning researchers have predicted attacks using statistical modeling methods, including GARCH models. For instance, the authors of [19] have characterized the honeypot captured data using statistical approaches. The authors have pointed out the importance of explainable statistical approaches for efficiently handling the prediction problems in honeypot data using case studies. The same authors have additionally predicted cyber-attack rates using GARCH prediction models in their following works [30]. Obviously, robust prediction models are crucial for proactively identifying the potential hackers or malicious attackers in modern networking applications, including IIoT or cloud services.

Apart from the normal prediction approaches which predict the potential hackers or their activities, a few researchers have devised honeypot mechanisms to protect against vulnerabilities arising out of the adversarial learning processes. Authors of [29] have suggested learning models that protect against adversarial errors opted by automated machine learning algorithms. For instance, IIoT applications, guided by machine learning services, could be exposed to wrong learning advice which could end up with hazardous results. To override such effects, honeypots were utilized to protect failures and rectify prediction failures.

As observed, there exists a few research works that utilize machine learning algorithms and mechanisms, including the Cloud services domain, for predicting or characterizing hackers. However, there are very few works that utilize explainable AI for validating the importance of honeypot predictions levied by machines or computing domains in an organization. It could be noticed from the literature that explainable AI has emerged in the recent past to justify the blackbox prediction approaches or prediction algorithms.

2.3 Blockchains for Honeypots

Although attackers and associated vulnerabilities could be predicted, the findings need to be protected. Insider attacks in most organizations have been highly dangerous due to the modifications and corrections held to the findings by potential inner-organizational hackers. Blockchains could protect against tampering with data in such environments. In addition, the security policies could vary depending on regional/organizational policies. For instance, blacklisting certain IP addresses depends on several factors, including the organizational relationships with associated countries.

There exists a few research works of honeypots relating to blockchains. However, they were applied in a different context. For instance, researchers of [25] and [5] have established data science algorithms to learn the potential fraudulent activities such as fraud payments due to the Ethereum smart contracts.

A very few research works have applied the permissioned blockchains to quickly validate the blacklisting IPs that reach honeypots.

This work endeavored to apply prediction models, such as RFM, LRM, and SVM to predict the potential hacking IPs and commit the information into the permissioned blockchain ledger. The entry of information into the ledger is governed by a few approvers, including the Explainer-AI of the proposed framework (see Section 3).

3 `EA-POT` Framework

Honeypots have been reasonably deployed alongside production systems in recent years to study the behavior of potential cyberattackers. Accordingly, the honeypots pave way for the security team of organizations to protect their systems from several vulnerable attacks. In fact, the potential attackers should be predicted in an explainable manner before the information were listed in an immutable database.

This section explains the inner details of the proposed `EA-POT` framework for blacklisting potential cyber attackers using explainable prediction models and blockchains.

The proposed `EA-POT` framework consists of the following entities:

- Honeypot Data Engine,
- Prediction Models,
- Explainable AI Components,
- Policy Stakeholders,
- Blokchain Network, and,
- `BlackBlock` Database.

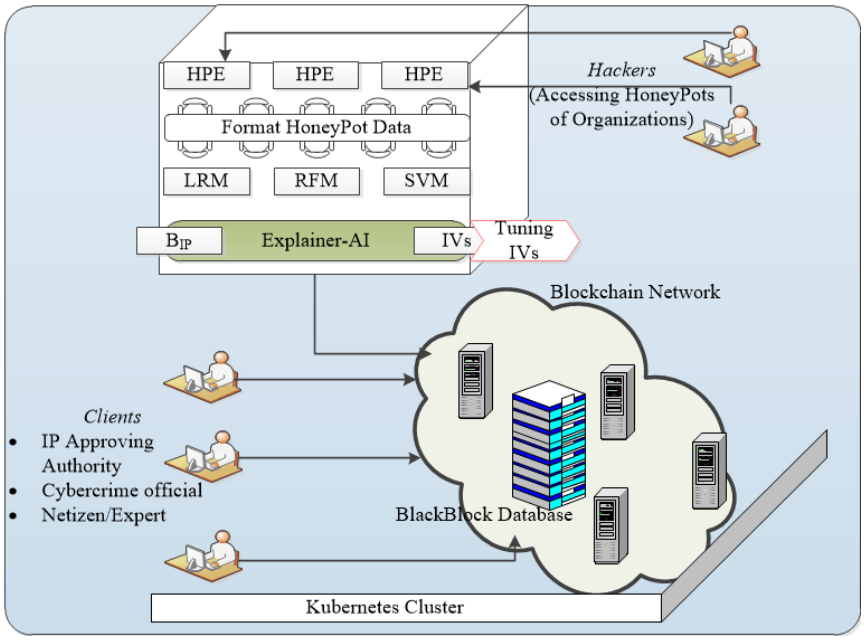


Figure 1: `EA-POT` Framework

The crucial functionalities of these entities are explained below.

3.1 Honeypot Data Engine

The *Honeypot Data Engine* is an entity that resides on honeypots that are located nearer to the firewall component of the organizations. It collects information, such as IP addresses, source port address, destination port addresses, connection protocols, such as TCP or UDP, country of origin, and so forth, of defaulters. Besides, it formats the information into CSV, XML, and JSON formats, in a periodic manner and keeps them ready for further processing of the intended prediction models of the framework.

3.2 Prediction Models

The framework utilizes a few notable algorithms, such as RFM, LRM, and SVM for predicting the potential hackers and their IP addresses. One battle in which the traditional honeypot engines allied to defeat progress was the timely identification of potential hackers' IP addresses. In doing so, several countermeasures could be adopted for overriding the issues.

The synopsis of the three prediction algorithms applied in the `EA-POT` framework is given in the following paragraphs.

Random Forest Modeling (RFM) Random Forest Modeling (RFM), the concept initially conceptualized by Breiman et al. [40], has been widely applied for creating prediction models that resemble real-world situations. It is an ensemble-based learning approach that creates decision forests based on modeling features. The models are created for the dependent variable of the dataset. For instance, the independent variable for honeypot IP prediction includes the IP addresses of potential hackers.

The decision forests consist of tens of hundreds of decision trees that analogously represent rules and inferences. Based on the creation of decision forests considering the decision rules for training data, the predictions are applied to the testing data. During the process of predictions, in the case of honeypot IP predictions, the independent variables, such as source port addresses, destination port addresses, latitude and longitude of locations, and so forth, are considered as modeling features – i.e., the independent variables.

RFM-specific tuning parameters[41], such as number of trees to grow in a forest (`ntree`), number of trials (`mtry`), and so forth, define the prediction accuracy on the testing dataset. For instance, increasing the number of trees could improve the prediction accuracy on large datasets.

Support Vector Machine (SVM) Support Vector Machine (SVM) is a supervised learning algorithm [11], as similar to RFM, where it attempts to produce hyperplanes that split data with sufficient distinctions. It attempts to increase the decision boundary of categorizing training data so that predictions could be much easier. The accuracy of the prediction algorithm is highly dependent on the dataset that is utilized – i.e., if the algorithm could not find sufficient hyperplanes, the error rate for predictions is typically higher than the expected ones.

During the training processes, the SVM algorithm iteratively prepares hyperplanes based on the independent variables of the dataset. To do so, it utilizes kernels, such as linear, polynomial, radial, and sigmoid, to transform the training data to a high dimensional space so that the process of creating hyperplanes is comparatively carried out elegantly.

Linear Regression Modeling (LRM) Linear Regression Modeling (LRM) is considered to be the simplest prediction model that identifies the relationship between the dependent and independent variables of a dataset. It highlights the potential changes that could happen in the dependent variable while modifying the independent variables. Not all independent variables are inclined towards the dependent variables of a dataset.

During the training processes of the linear regression algorithm, linear equations or mathematical formulas are created for the dependent variable based on the training dataset. In the proposed work, ML algorithms, such as RFM, SVM, and LRM are sufficient for learning the blacklisted IPs as decisions on confirming them are governed by a few stakeholders of blockchain networks. Accordingly, the policies could be varied as specified in the blockchains and the predictions are faster than

learning algorithms, such as neural networks.

3.3 Explainable AI Components

The prediction models are reasoned using the explainable AI components augmented in the framework. The framework applies explainable AI components to explore the inference levied from the models. For instance, the framework feeds specific modeling parameters to understand the R^2 values of the models. The R^2 values determine the closeness of the model and the dependent variables. The framework iterates over the available independent variables to identify the best set of independent variables $IV_{1...n}$ which offer the best R^2 values.

3.4 Policy Stakeholders

The policies for registering an IP to blacklists and for releasing the IPs from the blacklists need to be guided/formulated by multiple stakeholders. For instance, email hackers, the IP addresses, and port numbers of hacking applications need to be blacklisted depending on genuine reasons. Notably, blacklisting IP addresses due to technical failures reduces the reputation of an organization. Hence, in `EA-POT` framework, an array of policy stakeholders are represented for validating the genuineness of blacklisting IP addresses. In addition, it involves the explainable AI features to evaluate the necessity of blacklisting an IP into the immutable database.

3.5 Blockchain Network

The policy stakeholders of the `EA-POT` framework are connected to each other using a P2P blockchain network. These policy stakeholders are responsible for running policies or chaincodes; and, to interpret the data on server components. These server components, mostly established as a docker farm, are connected to each other using the blockchain network.

3.6 BlackBlock Database

The potential blacklisted IP addresses that are predicted and validated using the blockchain stakeholders of the network are registered into the blockchain ledger of `EA-POT` framework named as *BlackBlock* database. The reason to set up a blockchain database to register blacklisted IPs into the ledger is to protect the vulnerability raised by potential hackers, mostly the vulnerability due to the inner threats by colleagues of the same organizations. Figure 1 depicts on the entities involved in the `EA-POT` framework.

4 Explainable AI and Predictions

The recent era of machine learning development, in various research domains, has seen a proliferation of prediction models which can often be classified as blackbox

mechanisms. At this juncture, the evolution of explainable AI concepts has improved the trust levied by researchers on blackbox models. This section explains the interpretability procedure of prediction models of the `EA-POT` framework.

In general, a blacklisting of IP addresses happens due to several reasons:

1. an execution of a malicious program in a machine, including sensor nodes;

2. varying policies of organizations, which protect against the utility of certain types of applications – for instance, a military organization does not permit access to unauthorized military services;

3. inappropriate content, such as illicit videos and images in the services; and

4. spying of services within intra- and inter-organizations.

Predicting the blacklisting of an IP address in `EA-POT` framework attempts to avoid threats and strengthens the firewall policies depending on the learning inferences. In addition to a normal prediction process, `EA-POT` framework applies explainable features of AI to bolster the accuracy of predictions.

There may be various reasons for the formidable range of issues and inaccuracies of prediction models in modern applications:

1. the learning parameters are not appropriately chosen;

2. the modeling algorithms learn almost all available data – i.e., the model is biased concerning the data;

3. the training datasets are comparatively low; and so forth.

Obviously, it is an impressive activity for the user to understand the reason for predicting the blacklisting IP, an independent variable B_{IP}, with a specific level of accuracy considering dependent variables $X_{i...n}$. `EA-POT` utilizes local independent variable information of models for collecting $X_{i...n}$ that influence the predictions.

The major advantages of including the explainable features of the model in the `EA-POT` framework are:

- the features of *Explainer-AI* reveal the level of confidence of prediction models in R^2 percent; and,

- they establish a set of permutations from the observation instances and highlight the inclination of dependent variables towards the independent variable.

The *Explainer-AI* identifies the best suitable modeling parameters based on the R^2 values of the prediction models. Accordingly, the algorithms impose the choice for registering IP addresses into the blockchain ledger.

5 Immutability of `BlackBlock` and Processes

In `EA-POT` framework, *BlackBlock* database is established to list the blacklisted IP addresses that are predicted to be registered into the honeypots of organizations. In this section, the formation of a blockchain network, *Blackblock* database, and the processes involved in ensuring immutability are discussed.

5.1 Blockchain Network

The *BlackBlock* database of `EA-POT` framework is a distributed ledger that is established in the nodes of a Kubernetes cluster. The Kubernetes cluster [39] is chosen for the scalability and reliability features of distributed ledgers.

In general, Kubernetes is an orchestration tool that manages the containerized workloads of applications. It is manifested that the performance of Kubernetes clusters is better than many other orchestration tools while executing the containerized applications on them [18].

In `EA-POT` framework, the stakeholders of blockchains are represented as docker instances, which are containerized instances. The inclusion of the Kubernetes cluster enables users to evaluate and modify the state of docker machines, typically, the peer nodes of blockchains in the network.

The policy stakeholders of the framework that are represented in the docker instances include:

- IP Approving Authority,
- Explainer-AI,
- Cybercrime official, and
- Netizen/Expert.

These stakeholders have provisions to interact with the docker instances through docker client instances (see Figure 1). The docker instances, which represent the stakeholders of the `EA-POT` framework, install and launch chaincodes, the policies, for understanding the inferences of explainable AI, and for manipulating the entry of IP addresses into the ledger.

The chaincodes of the framework are written in `golang` language. These chaincodes are responsible for implementing policies of stakeholders where the Explainer-AI or similar stakeholders could determine the approval of transactions – i.e., the registering of blacklisted IPs into the database. The chaincodes are instantiated, installed, packaged, and queried using specific commands as shown below:

```
peer chaincode install/instantiate/...
```

The *Blackblock* database is protected within a specific channel that has connections to the permissioned stakeholders. The channel configurations and associated

information are defined before starting the blockchain network. The channel is responsible for establishing a sub-network where peer nodes could share the database within the organizations.

The blockchain network of the `EA-POT` framework emphatically complies with promoting a trustless trust environment using the distributed docker instances. The network offers ledger services across the connected nodes. It enables the nodes to readily keep the database for querying or modifying or manipulating the records in the database.

5.2 *BlackBlock* Database

The proposed `EA-POT` framework has a specific data structure to append blacklisted IP addresses into the *BlackBlock* database. The data in the database is appended as backlinked listed blocks for every initiation of transactions by peer nodes. The blocks are identified by hashes which include the previous hash values of the blockchain and the state of the blocks [37].

Each block is appended with a data structure that includes IP addresses, source port addresses, destination port addresses, and country information. Typically, the chaincode policies determine the entry of the blacklisted IP into the *BlackBlock* database.

The data appended into the database is sequential and immutable. The moment an entry would be registered into the ledger in a channel, the data will be visible to all available peer nodes of the channel.

5.3 Processes Involved

The processes involved in the entire life cycle of the `EA-POT` framework for registering the blacklisted IP addresses into the permissioned blockchain are described as steps below:

1. *Initiation:* In this step, the honeypot data engine and blockchain networks are initiated on top of the Kubernetes cluster. This means that the services are enabled at servers to attract potential hackers. In addition, the channel and peer networks are activated for implementing chaincode policies.

2. *Predictions:* Based on the available data, the learning models are created using sophisticated algorithms, such as RFM, LRM, and SVM. The generated regression models are utilized for predicting the future potential hacker IP addresses.

3. *Explanations:* Using the generated prediction models, explanations are developed using the independent variables of the models in the EA-BOT framework. The explanations are linked to the chaincode policies of the policy stakeholders of the blockchain network such that the stakeholders govern the control of blockchain transactions, including Explainer-AI.

4. *Chaincode Instantiation:* The stakeholders of the `EA-POT` framework are responsible for collectively agreeing upon entering the predicted IPs to blacklist them. Chaincode policies are defined in the `EA-POT` framework such that the stakeholders are diverse in nature – i.e., one stakeholder is *Explainer-AI*. This stakeholder evaluates the model that manifests a higher threshold of agreement while blacklisting IPs; another stakeholder is a representative of country authorities who approves and disapproves the blacklisted IPs – i.e., *IP Approving Authority*. This stakeholder evaluates the IPs concerning the country-wide policies set up for IPs; the other stakeholder is a netizen/expert who has a wide experience in executing a similar kind of applications and have the knowledge to judge the genuineness of actions to some confidence level. This stakeholder is named as *Cybercrime Official*, and the last stakeholder of the `EA-POT` framework is responsible for evaluating the IPs based on the genuineness of country-specific information.

5. *Transactions:* Once when the stakeholders agree on the possibility of the vulnerability of an IP address impacted on honeypots, the transaction to blacklist the IP address as an entry to the *BlackBlock* database is initiated by the orderer service of the hyperledger fabric-based permissioned blockchain [38]. Figure 2 illustrates the processes involved in the `EA-POT` framework in a pictorial form.

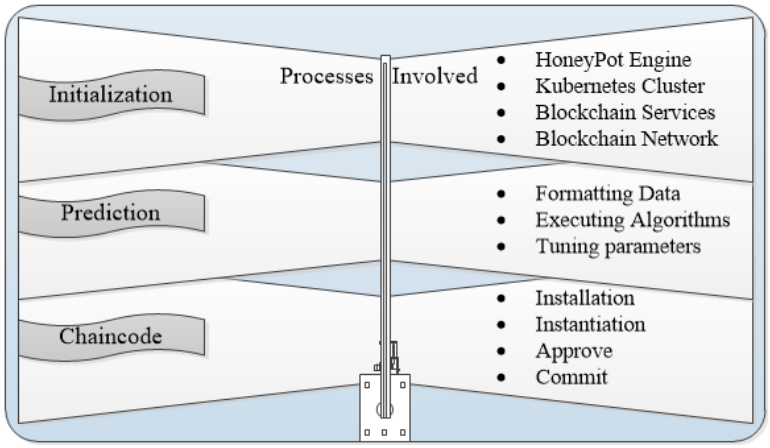


Figure 2: Processes Involved in the `EA-POT` Framework

6 Experimental Results

This section explains the experiments held at the IoT Cloud research laboratory. At first, the experimental setup is explained; next, the validation and prediction results

of identifying the potential IP addresses of hackers using prediction algorithms are explored; next, the application of explainable AI concepts, while including them in blockchains, for approving the transactions is discussed; and, at last, the entry of predicted IPs into the `BlockBlack` immutable database is showcased.

6.1 Experimental Setup

To mimic the scenario of receiving IP addresses into the honeypot engine of the `EA-POT` framework, AWS honeypot dataset [33] was utilized in the experiments. The honeypot dataset had 451581 rows of data with information, such as hacker IP addresses, country of origin, source port address, destination port address, latitude and longitude of the hacker, postal code, protocol, and date/time of the incident. Although any honeypot dataset could be applied for predicting potential hackers, in this work, the AWS honeypot dataset was utilized for the prediction models RFM, LRM, and SVM, to reveal the capability of the framework.

All experiments were carried out on four machines of IoT cloud research laboratory – i) a DELL precision tower 7810 machine which consists of 48 CPUs. This node serves as the master node of the Kubernetes cluster; and, ii) three i7 processor machines which serve as the worker node of the cluster. These nodes were interconnected based on the *Calico* networking policies [34] of the Kubernetes cluster.

On top of the Kubernetes cluster, a hyperledger-based permissioned blockchain was set up with the following configurations: fabric v2.0, dockerv19.03, and `golang` version 1.14. Four docker instances were established that represent the policy stakeholders of `EA-POT`, such as:

```
explainer-ai.com,
ip-approve-authority.com,
cybercrime-aiciiit.com, and
netizen.com.
```

The blockchain network was established using these docker machines that represent the organizations. Each organization had one peer for installing, instantiating, and executing the chaincode policies; the blockchain network had one channel to hold the blockchain ledger consisting of honeypot IPs; the peer of the `cybercrime-aiciiit.com` served as the `orderer` of the permissioned blockchain setup of the `EA-POT` framework.

For providing predictions, algorithms, namely, RFM, LRM, and SVM were written using R version 4.0.0. The prediction algorithms utilized 50 percent training data and the other 50 percent testing data during the validation processes.

6.2 Honeypot Data – Validation of Algorithms

Analyzing honeypot data of AWS using prediction algorithms, such as RFM, LRM, and SVM of the `EA-POT` framework could provide a better insight into the efficiency of the algorithms. Hence, the validation of subsets of data was analyzed. Figure 3

reveals the R^2 values of the prediction results. For RFM experiments, the number of trees was chosen as 100 (ntree=100) and the number of variables sampled at each split was chosen as 2 (i.e., ntry=2). For SVM experiments, the kernel was fixed as "linear"; the coefficient value was fixed as "0"; cache memory was chosen as 40 MB; the tolerance of termination criterion was chosen as 0.001; and, the epsilon value was fixed as 0.1. Additionally, the model was allowed to undergo probability predictions. For LRM experiments, the model type was chosen as "responsive". All prediction experiments were carried out such that the variable "ipnumber" of the dataset was chosen as the dependent variable; and, the independent variables were considered as "country code", "source port address", and "destination port address".

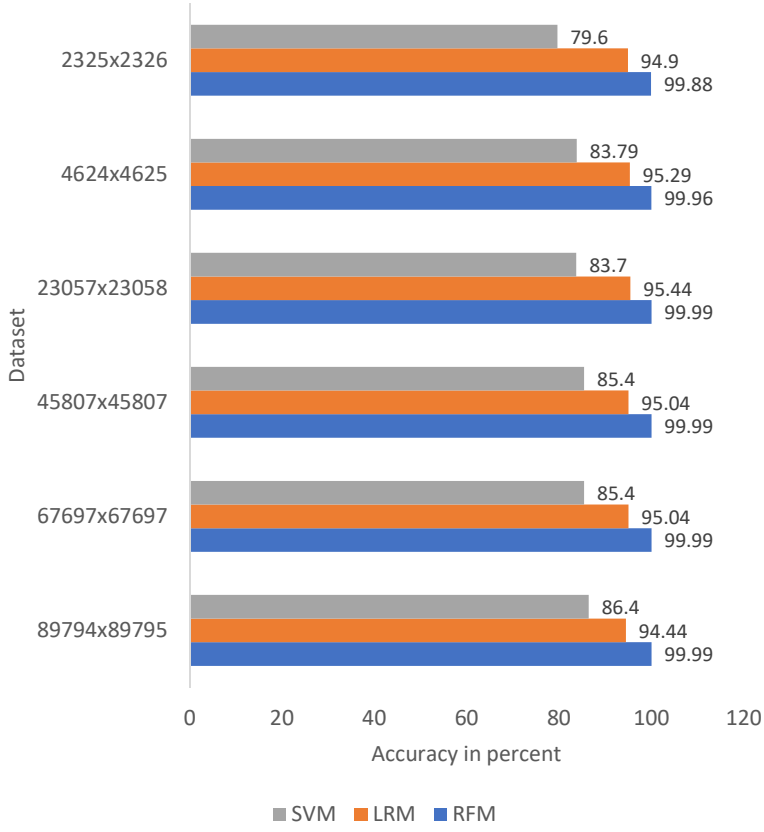


Figure 3: R^2 Values of RFM, LRM, and SVM

The following points could be observed from the Figure 3:

1. RFM algorithm performs well when compared to the other two algorithms of consideration. It could be observed that RFM has achieved around 99.99 percent accuracy when compared to the 85.4 percent accuracy of SVM.

2. Similarly, the prediction algorithm performs better when the training data size increases. For instance, the R^2 value of the SVM algorithm improved from 79.6 percent to 86.4 percent when the data size was increased from 2326 to 89795.

In addition, experiments were performed to study the variation of the prediction accuracy (R^2) while choosing different parameters in modeling algorithms. For instance, the R^2 value of SVM was reduced to 76.4 when the SVM modeling algorithm was executed with kernel="radial", coefficient=0, tolerance=0.01, epsilon = 1, and the probability of prediction was set to `TRUE`.

The time required for predicting these algorithms increased for a certain subset of analysis data. Table 1 illustrates the time required for processing data TDP, time for modeling data TM, and time for predicting data TP.

Table 1: Time Measured in Seconds For Data Processing, Modeling, and Prediction

Dataset	**Algorithm**	**TDP**	**TM**	**TP**
2325x2326	RFM	2.52	0.76	0.02
	LRM	2.42	0.01	0.002
	SVM	2.33	1.3	0.08
4624x4625	RFM	2.76	2.12	0.04
	LRM	2.45	0.017	0.0014
	SVM	2.37	4.95	0.32
23057x23058	RFM	3	3.75	0.23
	LRM	2.66	0.04	0.002
	SVM	2.57	2.3	7.1
45807x45807	RFM	3.26	1.68	0.75
	LRM	3.05	0.211	0.01
	SVM	3.18	11.8	27.3
67697x67697	RFM	3.04	3.48	1.13
	LRM	3.14	0.19	0.04
	SVM	3.82	1.58	58.51
89794x89795	RFM	3.14	5.98	1.107
	LRM	3.13	0.32	0.004
	SVM	3.69	1.08	65.71

Table 1 pinpoints that the modeling time was dependent on the available dataset. Increasing the data size of the dataset had an increase in the modeling and prediction time – i.e., RFM algorithm required $TM = 0.76$ seconds and $TP = 0.02$ seconds for 2325 x 2326; whereas, the same algorithm took over $TM = 1.08$ seconds and $TP = 65.71$ seconds for 89794 x 89795.

Another feature that was observed from Table 1 is the increasing prediction time of SVM when compared to LRM or RFM. Note that the prediction time of SVM reached 65.71 seconds when compared to RFM of 1.107 seconds. The average data

processing time reached 3 seconds for all these prediction algorithms. The data processing involved loading data, initializing dependent and independent variables, and splitting the training and testing dataset of the AWS honeypot data.

In addition, it was observed that varying the parameters of modeling algorithms influenced the TM. For instance, the RFM algorithm showed an increasing modeling time when experimented with more number of splits in variables while constructing the random trees – i.e., TM reached 10.92 seconds when RFM was executed with *ntree=4* in contrary to $TM = 5.98sec.$ for *ntree=2* of 89794 x 89795 dataset (see Table 1).

6.3 Prediction Results

Having validated the model, the potential hacker IP addresses were predicted for the specific location using RFM, LRM, and SVM algorithms. The prediction results obtained for a few candidate locations, when experimented with the RFM algorithm, are shown in Table 2.

The prediction of potential IP addresses that fall prey to the honeypot engine of organizations was reported in Table 2 using RFM prediction algorithm. In fact, the other algorithms could also be reported as similar to RFM. However, the reason for choosing RFM is because of its higher prediction accuracy when compared to the other algorithms namely LRM and SVM.

As shown in Table 2, the potential IP addresses that could harm organizations, that reach the honeypots, were initially predicted as numbers. Later, the numeric IP addresses were converted to IP numbers based on the `iptools` utility of R programs.

Table 2: Prediction of IP Addresses of Honeypot using RFM

Sl.No	**Latitude**	**Longitude**	**IP Addresses**	**Country**
1	37.49	127.02	218.237.65.47	South Korea
2	40.45	-105.46	129.82.138.44	United States
3	52.35	4.9167	8.16.85.133	Netherlands
4	55.154	61.429	31.207.238.106	Russia
5	39.715	-75.5281	199.59.160.152	United States
6	31.8639	117.2808	25.9.68.20	China
7	37.4906	127.02	60.173.14.88	China

6.4 Explainability Analysis

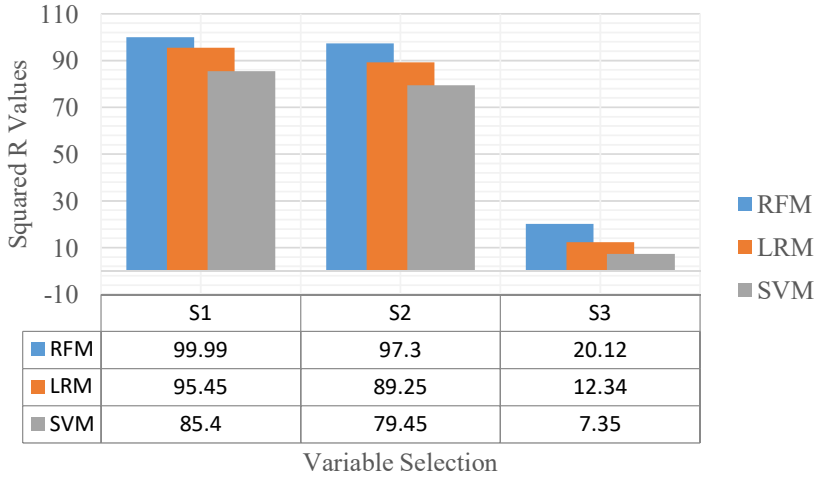
Explainability features of prediction algorithms reveal the prior importance of accurate predictions. The predictions carried out at `EA-POT` framework utilizes R^2 values to explain the importance of independent variables of prediction algorithms.

It is a known fact that most of the available models are considered black boxes – i.e., they may bestow better predictions without hinting at the reasons for achieving a better accuracy or without pointing out the most impacting independent variables for achieving the accuracy. In succinct, apt independent variables must be chosen for gaining better prediction results.

To manifest the influence of the choice of independent variables in the prediction results, experiments were carried out by varying the involvement of independent variables in the prediction processes of prediction algorithms.

In the experiments, three selection options $S1, S2$, and $S3$ were considered. The selections were organized to choose certain columns of the dataset – i.e., $S1$ utilized latitude, date, longitude, country code, source port, and destination port as independent variables while predicting the IP addresses; $S2$ utilized protocol, source port, and destination port addresses; and, $S3$ utilized all variables, such as date of occurrence, hostname, latitude, longitude, country code, source port, destination port, country name, and postal code for predicting the blacklisted IP addresses.

The explainability features of prediction algorithms were utilized by the blockchain chaincodes of the `EA-POT` framework. Figure 4 manifests that the variation in choosing inappropriate independent variables could lead to potential prediction inaccuracies – for instance, $S3$ has only 7.35 percent accuracy while predicting the IP addresses that reach the honeypots.



	S1	S2	S3
RFM	99.99	97.3	20.12
LRM	95.45	89.25	12.34
SVM	85.4	79.45	7.35

Figure 4: Variations in R^2 Values Depending on Independent Variables

6.5 Blockchain Transactions

It is not advisable to blindly choose the predicted IP address and protect the intended organizations or take countermeasures on the defaulters. Listing potential

IP addresses, therefore, needs to be diligently handled.

In `EA-POT` framework, permissioned blockchains using hyperledger fabric were applied. The master and worker nodes are set up such that docker instances representing the peer nodes of the blockchain network are executed on the Kubernetes cluster. During the experiments, the time taken to establish the Kubernetes cluster by the master node on three working nodes was 245.72 seconds. The clustering has several steps, such as creating the master node, joining worker nodes, deploying docker pods that represent the blockchain organizations, and specifying the domain names of the organizations for the fully operational cluster.

Once the predictions were carried out by the master node of the Kubernetes cluster, the predicted IPs are initiated as blockchain transactions by a peer node of the blockchain network. Note that all peer nodes install and instantiate the chaincodes – i.e., the policies for defining whether to register the IP address as blocklist into the *BlackBlock* database. In `EA-POT` framework, the organizations that approve the blockchain transactions are i) IP Approving Authority, ii) `EA-POT` Explainer, iii) CyberCrime Official, and iv) Netizen/Expert.

The time taken by the peer nodes of the blockchain network to install and query chaincodes was 13.35 seconds of which 12.78 seconds were spent on the installation of chaincode policies.

Predicting blacklisted IPs may not be successful at all times due to the accuracy of algorithms. Accordingly, it is not a good solution to blacklist all predicted IPs. Hence, in the proposed framework, the stakeholders of blockchains, based on the policies, decide to collectively agree on the IP addresses before they were registered in the immutable database.

To demonstrate the viability of choosing stakeholders for deciding the registry of IP addresses in the *BlackBlock* database, a few experiments reported in Table 2 were repeated. It was observed that all IP addresses that were predicted by the RFM algorithm in the experiments were not committed to the database – i.e., IP addresses "8.16.85.133" and "25.9.68.20" pointing to the latitude and longitude of countries, such as the Netherlands and China were incorrect. This is because a

Table 3: IP Addresses Committed to the `BlackBlock` Database

Table Entry	Approver 1 (IP Approving Authority)	Approver 2 (Explainer-AI)	Approver 3 (Cybercrime Official)	Approver 4 (Netizen Expert)	BlackBlock (Committed)
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓
3	✗	✓	✓	✓	✗
4	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓
6	✗	✓	✓	✓	✗
7	✓	✓	✓	✓	✓

few predictions could lead to wrong IP addresses when applied using prediction models. Accordingly, all policy stakeholders of the Blockchain network, namely the IP Approving Authority, did not approve the entry of registering the IP addresses to the *BlackBlock* database (see Table 3). Hence, only the IP addresses that were approved by all stakeholders were committed to the database.

Table 3 illustrates the records that were registered into the *BlackBlock* database. The database, being an immutable database, could not be modified by participants, including the intra-organizational participants. Thus, the proposed `EA-POT` framework achieves better efficiency in handling cybercrimes without any modifications to the predicted honeypot IP addresses.

7 Conclusion

The process of converting potential cyber threats into threat discoveries, learning, and ultimately developing security-enabled products, such as honeypots has been evidenced in recent years in various domains, such as IIoT and Cloud environments. Initial efforts to predict the potential hackers, either by establishing honeypots or the other cybersecurity features, predominantly save time and protect the limited compute resources from hackers, especially on cloud-based IoT services. Prediction approaches of the past indicate that blackbox prediction approaches were practiced with limited utility. Additionally, the hacker information was not well-protected, especially when the hacking was carried out within an organization by an insider employee.

This article proposed an Explainable AI-Assisted Blockchain Framework for honeypot IP predictions named `EA-POT` framework. The proposed framework applied explainable features of prediction models, such as Random Forest Modeling, Support Vector Machine, and Linear Regression Modeling, to approve the registry of predicted blacklisted IPs into the Blockchain database along with the other approvers, such as CyberCrime official of a country/region.

Experiments were carried out in the IoT Cloud research laboratory by establishing a hyperledger-fabric permissioned blockchain on top of the Kubernetes cluster consisting of four experimental compute nodes. The experiments manifested the efficiency of the proposed `EA-POT` framework using AWS honeypot use cases. The article explored the findings and reported how the `EA-POT` framework blacklisted potential IPs based on the policy stakeholders involving the explainable AI features of prediction models.

Acknowledgement

The author thanks AIC-IIITKottayam and BEL funding agencies for supporting this research work. In addition, the author thanks the reviewers and the editorial team of the journal for processing the article on time.

References

[1] Anwar, Ahmed H., Kamhoua, Charles, and Leslie, Nandi. Honeypot allocation over attack graphs in cyber deception games. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 502–506, 2020. DOI: 10.1109/ICNC47757.2020.9049764.

[2] Arivudainambi, D., Varun Kumar, K.A., Sibi Chakkaravarthy, S., and Visu, P. Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance. *Comput. Commun.*, 147(C):50–57, nov 2019. DOI: 10.1016/j.comcom.2019.08.003.

[3] Bauer, Johann, Goltz, Johannes, Mundt, Thomas, and Wiedenmann, Simeon. Honeypots for threat intelligence in building automation systems. In *2019 Computing, Communications and IoT Applications (ComComAp)*, pages 242–246, 2019. DOI: 10.1109/ComComAp46287.2019.9018776.

[4] Benedict, Shajulin. Energy efficient aspects of federated learning – mechanisms and opportunities. In Patel, Kanubhai K., Garg, Deepak, Patel, Atul, and Lingras, Pawan, editors, *Soft Computing and its Engineering Applications*, pages 38–51, Singapore, 2021. Springer Singapore. DOI: 10.1007/978-981-16-0708-0_4.

[5] Camino, Ramiro, Torres, Christof Ferreira, Baden, Mathis, and State, Radu. A data science approach for detecting honeypots in Ethereum. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9, 2020. DOI: 10.1109/ICBC48266.2020.9169396.

[6] Dodson, Michael, Beresford, Alastair R., and Vingaard, Mikael. Using global honeypot networks to detect targeted ICS attacks. In *2020 12th International Conference on Cyber Conflict (CyCon)*, Volume 1300, pages 275–291, 2020. DOI: 10.23919/CyCon49761.2020.9131734.

[7] Du, Miao and Wang, Kun. An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 16(1):648–657, 2020. DOI: 10.1109/TII.2019.2917912.

[8] Elijah, Olakunle, Rahman, Tharek Abdul, Orikumhi, Igbafe, Leow, Chee Yen, and Hindia, MHD Nour. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, 5(5):3758–3773, 2018. DOI: 10.1109/JIOT.2018.2844296.

[9] Foschini, Luca, Gavagna, Andrea, Martuscelli, Giuseppe, and Montanari, Rebecca. Hyperledger fabric blockchain: Chaincode performance analysis. In *2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020. DOI: 10.1109/ICC40277.2020.9149080.

[10] Hara, Kazuki, Sato, Teppei, Imamura, Mitsuyoshi, and Omote, Kazumasa. Profiling of malicious users using simple honeypots on the Ethereum blockchain network. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3, 2020. DOI: 10.1109/ICBC48266.2020.9169469.

[11] Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., and Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. DOI: 10.1109/5254.708428.

[12] Horák, Karel, Bošanský, Branislav, Tomášek, Petr, Kiekintveld, Christopher, and Kamhoua, Charles. Optimizing honeypot strategies against dynamic lateral movement using partially observable stochastic games. *Comput. Secur.*, 87(C), nov 2019. DOI: 10.1016/j.cose.2019.101579.

[13] Jafarian, Jafar Haadi and Niakanlahiji, Amirreza. Delivering honeypots as a service. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, pages 1835–1844, 2020. DOI: 10.24251/HICSS.2020.227, http://hdl.handle.net/10125/63966.

[14] Kostopoulos, Alexandros, Chochliouros, Ioannis P., Apostolopoulos, Thodoris, Patsakis, Constantinos, Tsatsanifos, George, Anastasiadis, Miltos, Guarino, Alessandro, and Tran, Bao. Realising Honeypot-as-a-Service for Smart Home solutions. In *Proceedings of the 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–6, 2020. DOI: 10.1109/SEEDA-CECNSM49515.2020.9221787.

[15] Lynda, Boukela, Zhang, Gongxuan, Bouzefrane, Samia, and Zhou, Junlong. An outlier ensemble for unsupervised anomaly detection in honeypots data. *Intelligent Data Analysis*, 24:743–758, 07 2020. DOI: 10.3233/IDA-194656.

[16] Mashima, Daisuke, Li, Yuan, and Chen, Binbin. Who's scanning our smart grid? Empirical study on honeypot data. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. DOI: 10.1109/GLOBECOM38437.2019.9013835.

[17] Mattei, Tobias A. Privacy, confidentiality, and security of health care information: Lessons from the recent WannaCry cyberattack. *World Neurosurgery*, pages 972–974, 2017. DOI: 10.1016/j.wneu.2017.06.104.

[18] Pereira Ferreira, Arnaldo and Sinnott, Richard. A performance evaluation of containers running on managed kubernetes services. In *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 199–208, 2019. DOI: 10.1109/CloudCom.2019.00038.

[19] Rajaboyevich, Gulomov Sherzod, Rustamovna, Salimova Husniya, and o'g'li, Ganiyev Asadullo Mahmud. Characterizing honeypot-captured cyber-attacks:

Statistical framework and case study. *International Journal of Innovative Analyses and Emerging Technology*, 2(5):63–67, May 2022.

[20] Rowe, Neil C., Nguyen, Thuy D., Kendrick, Marian M., Rucker, Zaki A., Hyun, Dahae, and Brown, Justin C. Creating effective industrial-control-system honeypots. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, pages 1845–1854, 2020. DOI: 10.33423/ajm.v20i2.3003.

[21] Saxena, Ms. Apurva, Ubnare, Gaurav, and Dubey, Anubha. Virtual public cloud model in honeypot for data security: A new technique. In *Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence*, ICCAI '19, page 66–71, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3330482.3330516.

[22] Shalaby, Salma, Abdellatif, Alaa Awad, Al-Ali, Abdulla, Mohamed, Amr, Erbad, Aiman, and Guizani, Mohsen. Performance evaluation of hyperledger fabric. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pages 608–613, 2020. DOI: 10.1109/ICIoT48696.2020.9089614.

[23] Surnin, Oleg, Hussain, Fatima, Hussain, Rasheed, Ostrovskaya, Svetlana, Polovinkin, Andrey, Lee, JooYoung, and Fernando, Xavier. Probabilistic estimation of honeypot detection in Internet of Things environment. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 191–196, 2019. DOI: 10.1109/ICCNC.2019.8685566.

[24] Tian, Wen, Ji, Xiaopeng, Liu, Weiwei, Liu, Guangjie, Zhai, Jiangtao, Dai, Yuewei, and Huang, Shuhua. Prospect theoretic study of honeypot defense against advanced persistent threats in power grid. *IEEE Access*, 8:64075–64085, 2020. DOI: 10.1109/ACCESS.2020.2984795.

[25] Torres, Christof Ferreira, Steichen, Mathis, and State, Radu. The art of the scam: Demystifying honeypots in Ethereum smart contracts. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 1591–1607, USA, 2019. USENIX Association. DOI: 10.5555/3361338.3361449.

[26] Vidal-González, Sergio, García-Rodríguez, Isaías, Aláiz-Moretón, Héctor, Benavides-Cuéllar, Carmen, Benítez-Andrades, José Alberto, García-Ordás, María Teresa, and Novais, Paulo. Analyzing IoT-based botnet malware activity with distributed low interaction honeypots. In Rocha, Álvaro, Adeli, Hojjat, Reis, Luís Paulo, Costanzo, Sandra, Orovic, Irena, and Moreira, Fernando, editors, *Trends and Innovations in Information Systems and Technologies*, pages 329–338, Cham, 2020. Springer International Publishing. DOI: 10.1007/978-3-030-45691-7_30.

[27] Wang, Binglai, Dou, Yu, Sang, Yafei, Zhang, Yongzheng, and Huang, Ji. IoTCMal: Towards a hybrid IoT honeypot for capturing and analyzing malware. In

2020 IEEE International Conference on Communications (ICC), pages 1–7, 2020. DOI: 10.1109/ICC40277.2020.9149314.

[28] Ye, Dongdong, Yu, Rong, Pan, Miao, and Han, Zhu. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*, 8:23920–23935, 2020. DOI: 10.1109/ACCESS.2020.2968399.

[29] Younis, Fadi and Miri, Ali. Using honeypots in a decentralized framework to defend against adversarial machine-learning attacks. In Zhou, Jianying, Deng, Robert, Li, Zhou, Majumdar, Suryadipta, Meng, Weizhi, Wang, Lingyu, and Zhang, Kehuan, editors, *Applied Cryptography and Network Security Workshops*, pages 24–48, Cham, 2019. Springer International Publishing. DOI: 10.1007/978-3-030-29729-9_2.

[30] Zhan, Zhenxin, Xu, Maochao, and Xu, Shouhuai. Predicting cyber attack rates with extreme values. *IEEE Transactions on Information Forensics and Security*, 10(8):1666–1677, 2015. DOI: 10.1109/TIFS.2015.2422261.

[31] Zhang, Weizhe, Zhang, Bin, Zhou, Ying, He, Hui, and Ding, Zeyu. An IoT honeynet based on multiport honeypots for capturing IoT attacks. *IEEE Internet of Things Journal*, 7(5):3991–3999, 2020. DOI: 10.1109/JIOT.2019.2956173.

[32] Australian cybersecurity expenditure. https://www.austcyber.com/resources/sector-competitiveness-plan-2019/chapter1, accessed in Oct. 2022.

[33] AWS honeypot data. https://www.kaggle.com/casimian2000/aws-honeypot-attack-data, accessed in Oct. 2022.

[34] Calico networking. https://docs.projectcalico.org/getting-started/kubernetes/self-managed-onprem/onpremises, accessed in Oct. 2022.

[35] Centrify forecasts. https://www.centrify.com/about-us/news/press-releases/2020/remote-working-increased-risk-cyber-breach/, accessed in Oct. 2022.

[36] Cloud security forecast. https://www.darkreading.com/cloud/cloud-security-spend-set-to-reach-%24126b-by-2023/d/d-id/1334473, accessed in Oct. 2022.

[37] Hyperledger block data structure. https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger.html, accessed in Oct. 2022.

[38] Hyperledger fabric. https://hyperledger-fabric.readthedocs.io/en/release-2.2/getting_started.html, accessed in Oct. 2022.

[39] Kubernetes cluster architecture. https://kubernetes.io/docs/concepts/architecture/, accessed in Oct. 2022.

[40] Manual on setting up, using, and understanding random forests v3.1. https://www.stat.berkeley.edu/~breiman/random-forests.pdf, accessed in Oct. 2022.

[41] Randomforest in R. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf, accessed in Oct. 2022.

Received 1st June 2021

Acta Cybernetica **26** (2023) 175–213.

Adding Semantics to Measurements: Ontology-Guided, Systematic Performance Analysis

Attila Klenik[ab] and András Pataricza[ac]

Abstract

The design and operation of modern software systems exhibit a shift towards virtualization, containerization and service-based orchestration. Performance capacity engineering and resource utilization tuning become priority requirements in such environments.

Measurement-based performance evaluation is the cornerstone of capacity engineering and designing for performance. Moreover, the increasing complexity of systems necessitates rigorous performance analysis approaches. However, empirical performance analysis lacks sophisticated model-based support similar to the functional design of the system.

The paper proposes an ontology-based approach for facilitating and guiding the empirical evaluation throughout its various steps. Hyperledger Fabric (HLF), an open-source blockchain platform by the Linux Foundation, is modelled and evaluated as a pilot example of the approach, using the standard TPC-C performance benchmark workload.

Keywords: performance, measurement, bottleneck identification, EDA, ontology, blockchain, Hyperledger Fabric, TPC-C

1 Introduction

The rapidly increasing number of IT service customers made the performance of such systems a high priority. Performant systems are not just a question of powerful hardware anymore, they also require the system-wide careful design of the software stack. The systematic detection and diagnosis of performance bottlenecks by analysing multi-dimensional measurement data becomes an integrated part of both the development and operational (DevOps) parts of the system life-cycle.

The industrialization of general-purpose data analysis resulted in typical standard workflows, like CRISP-DM [59], or ASUM-DM [11]. Such workflows are typically centered around the following high-level, domain-agnostic steps [6, 20, 52]:

[a]Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary
[b]E-mail: attila.klenik@vik.bme.hu, ORCID: 0000-0003-2051-2823
[c]E-mail: pataricza.andras@vik.bme.hu, ORCID: 0000-0002-6516-129X

DOI: 10.14232/actacyb.295182

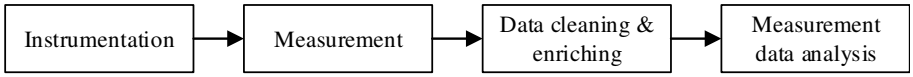


Figure 1: Typical performance evaluation steps

data acquisition; *representation*; *analysis*; *visualization and reporting* – with a proper model-driven engineering (MDE) support. Different performance analysis tasks – such as bottleneck identification and latency anomaly root cause analysis – can be considered domain-specific refinements [28] of the analysis step, defining further, embedded sub-workflows.

However, the *technical metrology* of performance evaluation poses specific challenges. The technical systems under test (SUT) are usually very complex, both in the terms of their architecture and potential state space. Still, performance engineering became increasingly important, as many systems have to fulfill soft real-time requirements. Moreover, poor performance dimensioning (stemming from architectural design or misconfiguration) can lead to service-level violations, or the malfunctioning of the system, even in the case of short overloads.

The paper proposes an *activity and observability-focused ontological approach* for the model-based guidance of SUT- and measurement-related, technical performance analysis tasks (Fig. 1): instrumentation; measurement; data cleaning and enriching; and measurement data analysis.

Instrumentation, the insertion of sensors into the system, plays an important role in system *observability*, i.e., the degree to which an observer can reconstruct the internal state of a system based on its outputs. However, sensor placement must balance multiple requirements: non-intrusivity whenever possible; development time/cost; and sufficient amount of resulting measurement data to work with.

On one hand, increasing the number of sensors might provide a deeper insight into the system, but application-specific sensors require a careful development to assure the integrity of the measurement results without distorting temporal metrics. On the other hand, under-instrumentation confines the granularity of root cause analysis and consequently the indication for mitigating bottlenecks. Moreover, it can leave faulty behavior undetected.

Correspondingly, instrumentation needs a careful trade-off between the relevance and redundancy of the measurements. The proposed approach aids the designer or analyst in formally arguing about the observability of the system, or in selecting a sufficient sensor placement.

During the *measurement*, data acquisition has to cope with the heterogeneity of data sources generating observation logs in very different formats. Data source models [27, 52] support the semantic fusion and representational homogenization of the sources and the following ETL (extract, transformation, load) steps.

The proposed ontology guides the ETL process towards a *relation-oriented and activity-focused representation* of measurement data, building on widely used concepts. The common format may serve as a gateway toward other temporal modeling

frameworks (e.g. the OWL Time ontology[1] from the World Wide Web Consortium), metrology-related technologies (e.g., the OpenTelemetry[2] project from the Cloud Native Computing Foundation), or other analysis techniques (such as process mining [56]).

Clean and detailed data is a prerequisite for many performance analysis tasks, such as bottleneck identification. Large-scale system observations constitute as *big data*, but more importantly, as *multi- or many-dimensional data*. Data is harvested from multiple layers of numerous system components, ranging from boundary-level response times to infrastructure-level resource utilization metrics. Bottleneck identification in such a context is a complex diagnostic process. It is a priori unknown how deep the analysis must drill down to uncover root causes of performance anomalies.

The proposed approach makes data validation a *systematic process* by sharding and inspecting the data set based on the modeled activities and corresponding services. Thus data omission errors, for example, can be easily pinpointed even in larger data sets. Moreover, the activity and observability models coupled with various temporal rules provide a framework for automatically *deriving further temporal information*, even if not explicitly observed.

The *analysis* of the gathered multi-dimensional data necessitates *exploratory data analysis* (EDA). EDA is, by its nature, a highly adaptive and iterative process for identifying a model of the system behavior. Usually, a domain expert is in charge of guiding the *drill-down process* if something peculiar is detected from the point of view of the application. The exhaustiveness and quality of this exploration process heavily depend on the domain knowledge of the expert, the automation of the elementary steps, and a proper navigation along the process and the data [52, 14, 45, 63].

The *hierarchical nature* of the proposed activity ontology and the corresponding service/deployment information make the drill-down process intuitive and systematic. The domain expert can dissect higher-level activities as needed, until a possible cause is found for a peculiar behavior. Then they can correlate the time range of the behaviour with various workload metadata and/or resource utilization metrics to find its root cause (may it be a bottleneck of the system, or a resource saturation issue). Furthermore, the drill-down steps are guided by concepts independent of the actual SUT, making the process reusable for different systems, a viable candidate for automation, or to be performed by a less experienced domain expert.

A complex case study demonstrates the benefits of the proposed approach:

- The *HLF blockchain* platform's consensus *activities* and their *observability* are modelled in a reusable and modular way.
- The activities of the standard TPC-C performance benchmark[3] are modeled and combined with the HLF model.

[1] https://www.w3.org/TR/owl-time/
[2] https://opentelemetry.io/
[3] http://tpc.org/tpcc/default5.asp

- The formal measurement inference capabilities of the ontology are demonstrated, coupled with a systematic data validation process.
- An ontology-guided, EDA-based, hierarchical bottleneck identification process is demonstrated on measurement data observed while executing the TPC-C workload (generated by the Hyperledger Caliper[4] benchmarking tool) on HLF.

The paper is structured as follows. Sec. 2 introduces the proposed general approach for the performance analysis of complex systems. Sec. 3 presents related MDE approaches for activity modeling and surveys the state of the art HLF performance researches. Sec. 4 introduces the elements of the proposed ODK used for complex activity modeling and automatic observability inference, along with formal semantics. Sec. 5 presents the case study of compositional modeling of the HLF consensus process and the TPC-C benchmark execution using the ODK. Sec. 6 demonstrates the various applicability of the resulting system models in aiding complex measurement data validation and analysis tasks. Sec. 7 concludes the paper.

2 The proposed model-guided analysis approach

The cornerstone of a performance analysis process is the observability of the activities performed by a system. On a high level, the beginning, duration, and end of system tasks are the basis of common metrics, like incoming task rate, throughput, and latency. Bottleneck identification, however, requires more interconnected data to work with, including the well-defined composition semantic of complex activities. Moreover, such observations are also crucial for building precise, well-parametrized models for efficient performance prediction [8, 15].

Our contribution is an ontology-guided workflow for the *systematic, drill-down performance analysis* of multi-dimensional measurement data. Moreover, the supporting *ontology development kit* (ODK) is provided for ensuring the quality and sufficiency of measurement data, enhanced with composition semantics for facilitating bottleneck identification processes. The ODK supports the modeling of activities, their relations, and whether their execution is observable outside of the system. Furthermore, it provides a formal foundation for rigorous measurement data analysis task, e.g., bottleneck identification.

The proposed approach is outlined in Fig. 2 and detailed in subsequent sections:

1. Model the important activities of the system components, focusing on their relations and hierarchical composition.
2. Model the explicit observability of activities to assist observability inference.
3. Extend the model with additional elements (by bridging to other ontologies, for example) to support further design, DevOps, or analysis tasks, as needed.

[4] https://www.hyperledger.org/use/caliper

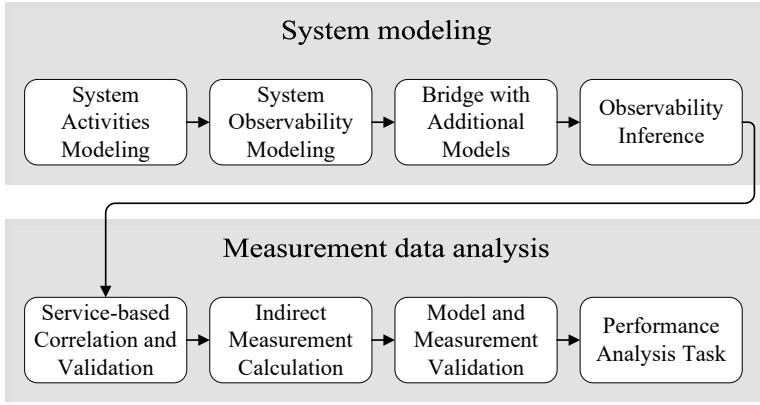


Figure 2: The proposed workflow for model-guided performance evaluation

4. Automatically enrich the current model with additional observability information by using an OWL reasoner (detailed in Sec. 4.5).

5. Correlate and validate distributed request traces from the SUT to uncover data omission or similar errors.

6. Calculate additional, indirect measurement using the ontology (or other derived) model as guide.

7. Validate the conformance of the measurement data and the model to ensure the correctness of further analysis tasks.

8. Perform the desired analysis task based on rigorously cleaned and validated data, and using the ontology model as guide.

Following the outlined steps allows for a rigorous performance analysis of the SUT. Note that the model construction steps only need to be performed once (if done properly), then the component models can be reused and recombined to fit further performance analysis scenarios for different SUT setups. Moreover, the modeling and analysis parts of the workflow can be performed by different domain experts, lowering the entry barrier for the overall performance analysis of a given SUT.

3 Related work

The section presents the related work on creating activity execution models and surveys the state of the art regarding HLF performance evaluations. The limitations of the presented literature motivated our contribution to bring MDE approaches closer to the domain of performance evaluation of complex systems.

3.1 Activity modelling

System activities are usually observed through individual events (e.g., logs, notifications) or sensors. An important requirement of activity modeling – relating to performance analysis – is to allow the systematic reconstruction of detailed timelines from the available partial observations, facilitating data analysis. Furthermore, having well-defined modeling semantics and building blocks allow the assessment of a wide range of systems.

Our experience with EDA and bottleneck identification outlined the following requirements for model-based support:

- *formal modeling* of complex *activity hierarchies* and relations;
- explicit modeling of system *observability* (i.e., sensor placement);
- *systematic derivation* of additional temporal knowledge;
- *extensibility* for incorporating further service/infrastructure models;
- *composability* and *reusability* of different activity models.

Similar approaches exist in the domain of business process analysis using online analytical processing (OLAP) [1, 39, 42, 48]. However, our approach has to comply with the additional requirements of technical metrology, like allowing the performance evaluation of general system activities despite limited observability of tasks, and facilitation of the adaptation of metrology principles.

Modeling the execution of activities also has a long tradition in software development, both as design phase artifacts for validation, and as inputs to automatic task orchestration systems. Business process models (building on the BPMN[5] standard) or activity diagrams in UML[6] or SysML[7] are prime examples of high-level activity modeling languages.

Such visual languages facilitate the modeling of activity control flows, imposing certain temporal constraints (e.g., activity A must be executed *before* activity B). However, the enforcement of such constraints must be validated during analysis time or runtime. Such validation necessitates the detailed observation of activities to allow rigorous temporal constraint checks. Moreover, the available high-level languages lack an intuitive support of modeling observability.

Additional formal approaches can also aid the *design* and *verification* of complex systems or processes. Temporal logics – such as Temporal Logic of Actions (TLA, TLA+ [30, 31]), Propositional Temporal Logic (PTL [46]), Interval Temporal Logic (ITL [3]) – enable the specification and verification of time-dependent system behavior [9]. Furthermore, different probabilistic or stochastic process algebra approaches [22] can aid the design and verification of concurrent, distributed systems, including the communication and synchronization of their independent

[5] https://www.omg.org/spec/BPMN/2.0/
[6] https://www.omg.org/spec/UML/
[7] https://www.omg.org/spec/SysML/

components. Such process algebras include the Performance Evaluation Process Algebra (PEPA [16]), or extended versions of the Communicating Sequential Processes (CSP [32]) language.

The above formal approaches provide an expressive power sufficient for the detailed design, verification and subsequent implementation of concurrent systems. However, their application requires knowledgeable experts in the formal verification domain. Moreover, the intended purpose of the presented approaches is to aid the correct design and implementation process of complex systems, thus providing more facilities than needed for validating and analyzing performance measurements of the already implemented systems.

Accordingly, the goal of the current work is to provide a kernel modelling framework (with simple vocabulary and relations pertaining to system activities) that is easier to use, given a standard information technology background. The semantic mapping of the referenced formal approaches (and the corresponding, readily available formal design models during system development) to the presented modelling framework is left as future work.

Ontology-like formal approaches also gain ground in general system modeling tasks (e.g., the upcoming OMG SysML v2 Kernel Modeling Language[8]), thus our contribution relies on ontologies, preparing for future interoperability. Knowledge representation-based approaches can also aid the visual analysis of network traffic [61] or the semantic fusion of data originating from different sources [60]. Moreover, ontology-based approaches can reason about the occurrence of composite activities [43, 12, 21, 33, 49].

The referenced activity modeling works have several elements in common. They utilize Allen's interval algebra [2] for describing temporal relations, allowing bridging to other similar solutions. However, they reverse-engineer/infer the activity model based on the observation of performed activities, similarly to process mining [56]. Model mining is unavoidable in contexts where the "schedule" of executed activities is non-deterministic, such as in smart homes or in smart warehouses [12, 49].

However, when the execution of activities must conform to a predefined specification, model mining becomes unnecessary. The paper proposes a *model-first* approach to construct an ontology-based composite activity model, which will later provide a strong foundation for the systematic performance evaluation and bottleneck analysis of the target system. Accordingly, the model becomes an input to the analysis tasks, and not an output.

3.2 Hyperledger Fabric performance analysis

The complex consensus process of HLF [5] (detailed and modeled in Sec. 5) made its performance evaluation a hot research topic. Related works can be divided mainly into the following categories based on their goals:

[8] https://github.com/Systems-Modeling/SysML-v2-Release

1. Performance *evaluation and characterization*: [47, 7, 55, 38, 18, 34, 51, 19, 58, 29, 40, 24, 41, 4, 13, 57, 50, 10]
2. Performance *optimization*: [55, 17, 25, 37]
3. Formal consensus *modelling*: [53, 54, 26, 64, 62]

Category 1 receives most of the attention, which is identifying the performance characteristics of HLF. The evaluations employ empirical sensitivity analyses to measure the change in key performance indicators (such as throughput and end-to-end latency) when applying different network scales, configurations, and workloads.

The concern of Category 2 is the performance enhancement of HLF. Researches either transparently optimize certain consensus steps or propose changes to the architecture (and correspondingly the consensus process) itself. The researches of Category 2 also rely on empirical performance analysis to confirm bottlenecks and evaluate the effectiveness of optimizations.

Works in Category 3 build formal behavior models of the consensus process. Model parameter identifications also rely on empirical performance evaluations. Finally, the parameterized model allows for cost-efficient sensitivity analyses capable of covering a large configuration and parameter space, without actual further empirical analyses.

Table 1 further refines Categories 1–3 based on the following aspects:

- *Network scaling* (NS): the research performed multiple evaluations while varying the scale of the Fabric network, e.g., the number of orderer and/or peer nodes (horizontal scaling), or their allocated resources (vertical scaling).
- *Configuration sweep* (CS): the Fabric network was evaluated in multiple operational modes, e.g., using different transaction ordering protocols (Kafka- or Raft-based implementations), varying the configuration of a specific ordering protocol (target block size or block time), or changing the implementation of other elements (the choice for state database, or the used chaincode language).
- *Workload scaling* (WS): certain attributes of the workload were changed among multiple evaluations, e.g., the incoming rate of requests, or the ratio of read and write intensive requests.
- *Consensus step optimization* (CSO): the research proposed improvements to certain steps of the consensus process that are transparent (i.e., non-breaking, backward compatible changes) to other network components or users, e.g., the parallelization of inner component tasks (like transaction validation).
- *Consensus process optimization* (CPO): the proposed improvements significantly change the overall consensus process, i.e., the changes require the adaptation of certain APIs, breaking the existing solutions.
- *Consensus modelling* (CM): the research formally modelled and evaluated certain aspects of the Fabric consensus, e.g., calculated expected transaction latencies based on the approximated processing times of subtasks.

Table 1: Categorization of Fabric performance evaluation-related works (**NS**: Network Scaling, **CS**: Configuration Sweep, **WS**: Workload Scaling, **CSO**: Consensus Step Optimization, **CPO**: Consensus Process Optimization, **CM**: Consensus Modelling)

Related Works	NS	CS	WS	CSO	CPO	CM
[55]	✓	✓	✓	✓		
[7, 57, 50]	✓	✓	✓			
[13]	✓	✓				
[38, 58, 40, 10]	✓		✓			
[34]	✓				✓	
[51]		✓	✓	✓		
[17]		✓	✓		✓	
[19, 37]		✓	✓			
[25]		✓		✓		
[4]			✓		✓	
[47, 18, 29, 24, 41]			✓			
[53, 54, 26, 64, 62]						✓

A common requirement for all categories and aspects is the rigorous empirical performance evaluation of HLF based on the analysis of measurement data. Superficial analyses may lead to incorrect hypotheses or misidentified model parameters, invalidating the results of the evaluation.

Accordingly, a *systematic*, *rigorous*, and *easy to follow* analysis process (even for complex systems) is needed to achieve relevant results. Moreover, the *correctness* and *richness* of measurement data can further increase the quality of gained insights.

4 Activity and observability modeling framework

The section introduces the formal foundations and building blocks of the proposed ODK for constructing complex activity models. Moreover, it details the observability modeling and automatic observability inference mechanisms that are the cornerstones of a rigorous performance data analysis.

4.1 Formal foundations

The ODK is constructed using the Web Ontology Language[9] (OWL2), adhering to some constraints (resulting in an OWL-DL ontology) that make the OWL direct semantics compatible with the model-theoretic semantics of the $\mathcal{SROIQ}$ description logic [23]. This restriction provides useful computational properties for the

[9] https://www.w3.org/TR/owl2-syntax/

language, backed by extensive literature and tooling support, such as OWL-DL reasoners [44].

OWL2 provides facilities such as object and data properties, literals, individuals, and classes to model relations among different concepts and resources. Classes can have associated relationship constraints that must hold for every individual belonging to the class. It is important to note that OWL employs the open-world assumption, meaning that if something is not asserted as knowledge, it is taken as unknown, rather than as untrue. The OWL2 structural specification[10] further details the available language constructs and their meaning.

The paper utilizes the OWL-DL notations of Table 2 to describe the elements of the ODK and their semantics.

Table 2: OWL-DL notations

OWL construct	Notation	OWL construct	Notation
$Class$	$C1, C2$	$SubClassOf$	$C1 \subseteq C2$
$IntersectionOf$	$C1 \cap C2$	$UnionOf$	$C1 \cup C2$
$Thing$	T	$Property$	P
$PropertyRange$	$T \subseteq \forall P.C1$	$EquivalentClass$	$C1 \equiv C2$
$AllValuesFrom$	$C1 \subseteq \forall P.C2$	$SomeValuesFrom$	$C1 \subseteq \exists P.C2$

The temporal constructs of the ODK build on Allen's interval algebra. Let us consider activity instances $a = (a^b, a^e, a^d) \in A$ of an activity class with a beginning time instant $a^b \in \mathbb{N}^+$, an ending time instant $a^e \in \mathbb{N}^+$, and a non-zero duration $a^d \in \mathbb{N}^+$, where $a^b < a^e$, and $a^b + a^d = a^e$ for every $a \in A$, measured on a logical clock for the simplicity of the notation.

If a property (i.e., directed relation) P holds between activity instances a and b, we denote it by $a \in P.b$, where $a \in A$, $b \in B$ activity classes. The shorthand notation $A \subseteq P.B$ specifies the relation P as constraint between activity classes A and B, implying $\forall a \in A : \exists b \in B, a \in P.b$.

The ODK defines the following Allen interval relations as OWL properties: $after$, $before$, $meets$, $metBy$, $starts$, $startedBy$, $finishes$, and $finishedBy$. Accordingly, if an activity type A is always followed by an activity type B, the axiom $A \subseteq meets.B$ will be part of the ontology. Note, that the ODK contains only the Allen relations that provide precise or useful activity composition semantics. Accordingly, the *during*, *overlaps*, and *equal* relations (and their inverses) are not utilized directly, but can be derived from the modeled relationships in a straightforward way.

[10] https://www.w3.org/TR/owl2-syntax/

4.2 Component overview

The ODK contains a hierarchy of smaller ontologies – each with well-defined responsibilities – to promote composability (Fig. 3).

The *Activity* ontology (Sec. 4.3) allows the modeling of system activity relations. For example, the following set of assertions partially describe an activity decomposition (Fig. 4): $Processing \subseteq SequentialActivity$, $Substep_i \subseteq AtomicActivity$, and $Processing \subseteq hasSubactivity.Substep_i$.

The *Observability* ontology (Sec. 4.4) provides classes to "annotate" the activities with further information regarding their degree of observability. For example, $Processing \subseteq EndMeasured$ denotes that the end of *Processing* activities are explicitly observed/measured through logs, or system events.

The *Structural constraints* ontology provides well-formedness axioms for activity composition. The open-world semantic of OWL2 makes it cumbersome to convey traditional (closed-world) modelling intentions to a set of ontology axioms. For example, it is not enough to just state that a subactivity is the starting activity of its parent. Correct modeling also requires the statement that the starting activity is not preceded by any other activity (otherwise it could not be the first subactivity of its parent).

The structural constraints ontology provides several axioms that can automatically detect (using an ontology reasoner) such inconsistencies or potentially missing axioms. The description of constraints, however, is outside the scope of this paper.

The *inference* ontologies (Sec. 4.5) extend the observability ontology with equivalence axioms that can automatically flag (during reasoning) activity classes

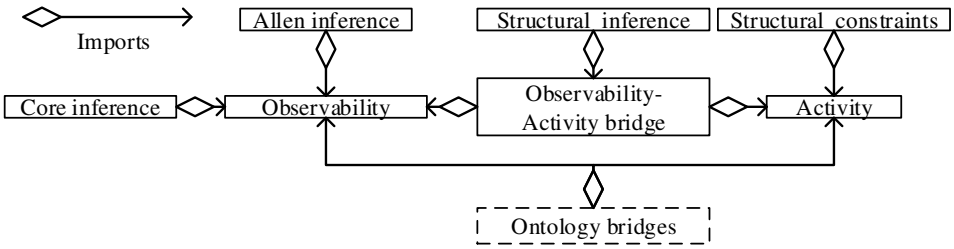


Figure 3: Ontologies in the ODK

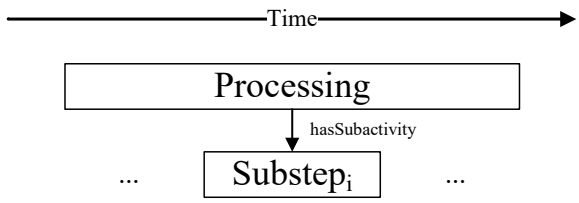


Figure 4: Example of a partial activity decomposition

based on their degree of observability. Such observability flags are propagated during reasoning along the activity relations, resulting in a complete observability description of the entire activity hierarchy.

The structural constraint and inference ontologies can be referred to as *aspect ontologies* in general: they separate orthogonal modelling concerns in a modular way and can be used to enrich a base model (similarly to aspects is aspect-oriented programming). Accordingly, a modeler can work using light-weigh and simple ontology concepts (activities and observability), and only perform possibly heavyweight computations/reasoning periodically by including the aspects.

The recommended modelling approach of the ODK is to decompose the complete system model into smaller ontologies (Fig. 5) for maximum flexibility and reusability.

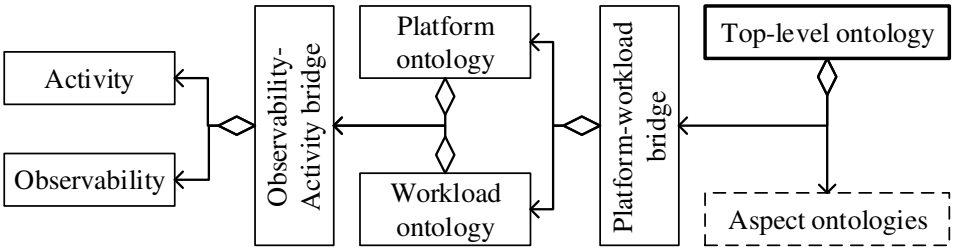


Figure 5: Ontology structure of a multi-component use case

A platform ontology models the composition of executed activity steps and their observability utilizing the ODK core vocabulary. A specific platform is usually just a means to execute a higher-level business scenario, which steps should be modeled as a platform-independent workload ontology whenever possible. This separation allows the flexible evaluation of different architectural/platform design choices by providing a platform-workload bridging ontology for specific scenarios.

The final element of the stack is a top-level (possibly automatically constructed) ontology that unites the pure system model with the chosen aspects of the ODK. Various OWL-DL reasoners can validate and enrich the top-level ontology, resulting in a detailed knowledge representation of the system that will serve as a basis for later performance analysis tasks. A concrete case study following the presented approach is detailed in Sec. 5 through modeling TPC-C benchmark execution on HLF networks.

4.3 Modeling activity hierarchies

Activity (*ACT*) hierarchies are defined with atomic ("leaf") elements and composite elements supporting further refinement (Fig. 6). The ODK provides the following *Activity* subclasses for modeling activity composition through subsumption relations:

- *AtomicActivity* (*AA*) represents elementary steps without further refinement;

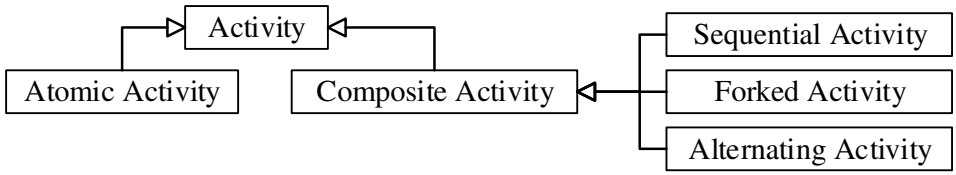


Figure 6: Hierarchy of activity types

- *CompositeActivity* (CA) allows further refinement of activities through the following subclasses, representing different composition semantics:
  - *SequentialActivity* (SA) allows refinement into a sequence of subactivities;
  - *ForkedActivity* (FA) allows refinement into parallel subactivities;
  - *AlternatingActivity* (TA) allows refinement into subactivities without additional control flow constraints.

The core classes on the same hierarchy level are disjoint ($AA \cap CA = \emptyset$, $SA \cap FA \cap TA = \emptyset$). However, uncategorized activities and additional composition semantics are allowed to promote extendability, i.e., $AA \cup CA \neq ACT$ and $SA \cup FA \cup TA \neq CA$.

The following high-level relations (OWL object properties) provide the basis for constructing complex hierarchies of activities:

- $Substep_i \subseteq hasParentActivity.Parent$, denoting that activity type $Substep_i$ has a parent (encapsulating) activity of type $Parent$.
- $Parent \subseteq hasSubactivity.Substep_i$, denoting that activity type $Parent$ has a subactivity (a refined substep) of type $Substep_i$.
- $Substep_i \subseteq hasSiblingActivity.Substep_j$, denoting that $Substep_i$ has the same parent activity type as $Substep_j$, i.e., $\exists Parent$ such that $Substep_i \subseteq hasParentActivity.Parent$, and $Substep_j \subseteq hasParentActivity.Parent$

The composite activity subclasses denote the typical (de)composition constructs for activity executions:

Sequential activities (SA) group together a sequence of subactivities that follow traditional sequential execution semantics. Moreover, refined relations are introduced (with Allen interval-like semantics, as mapped in Table 3) to further enrich parent-subactivity and sibling relations. The "synonyms" for the Allen relations were introduced to hint at the compositional nature of the activities (and not just their relative temporal placement), aiding modelers with traditional activity modeling backgrounds.

Note, that "gapped" relations indicate an incomplete timeline, hindering later analyses, and probably warranting additional instrumentation. However, the

Table 3: ODK and Allen relation mappings for sequential composition

Parent Relation $\supseteq$	Subrelation $\equiv$	Allen relation
$hasParentActivity$	$startsParentActivity$	$starts$
	$finishesParentActivity$	$finishes$
$hasSubactivity$	$startedBySubactivity$	$startedBy$
	$finishedBySubactivity$	$finishedBy$
$hasSiblingActivity$	$hasImmediatePredecessorActivity$	$metBy$
	$hasImmediateSuccessorActivity$	$meets$
	$hasGappedPredecessorActivity$	$after$
	$hasGappedSuccessorActivity$	$before$

ODK inference rules can be easily extended to detect "unknown", albeit observable activities whenever possible.

Forked activities (FA) group together parallel subactivities that are executed independently of each other. An associated synchronization/join semantic class ($\subseteq hasSyncSemantic.SyncSemantic$) can be used to model the condition when the parent activity is deemed finished.

The ODK defines two synchronization semantics: when *all* ($WaitForAll \subseteq SyncSemantic$), or *any* ($WaitForAny \subseteq SyncSemantic$) of the subactivities must finish to consider the parent activity done. Extending ontologies can define further semantics, e.g., waiting for the majority of subactivities.

Alternating activities (TA) are decomposed into a set of subactivities, disregarding control flow restrictions in cases when the control flow of subactivities is irrelevant. TA is a tool of abstraction for concentrating only on the "weight" (i.e., duration) of a subactivity, and not on its scheduling.

A typical use case is the modeling of in-process execution times and database access times of a task, disregarding execution semantics among the substeps: $Task \subseteq TA$, $InProcExec, DbAccess \subseteq hasParentActivity.Task$. Modeling the exact activity flow of computation and database access can be cumbersome for some use cases. Moreover, it may be sufficient during performance analysis to consider only the time/duration spent with each processing types, instead of focusing on their exact, possibly rapidly alternating order.

4.4 Modeling observability

Once the activity model is complete, the next step is modeling which activity temporal aspects (beginning, duration, and/or end) are measured directly in the system (i.e., modeling the placement of sensors and instrumentation) using the *Observability* ontology concepts.

The core concepts can be grouped into three main categories (Fig. 7):

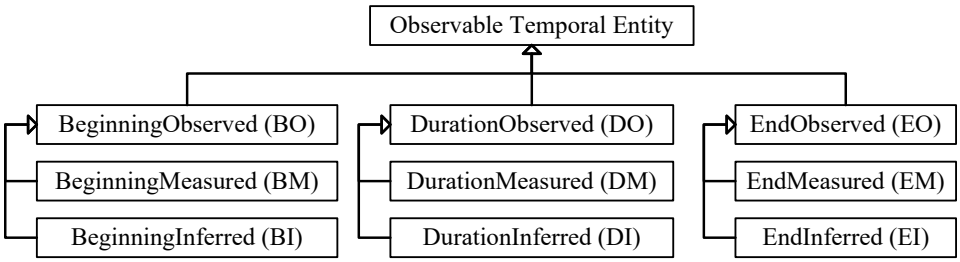


Figure 7: Observability ontology components, with abbreviations

1. observable data in the abstract sense (e.g., $Task \subseteq EndObserved$), denoting that the temporal data is available in some way (measured or inferred);
2. directly measured data (e.g., $Task \subseteq EndMeasured \subseteq EndObserved$), denoting that the data is explicitly measured;
3. and inferred data (e.g., $Task \subseteq Rule \subseteq EndInferred \subseteq EndObserved$).

The following class abbreviations are used in some places for readability (Fig. 7): BO, BM, BI, DO, DM, DI, EO, EM, and EI.

The modeler must "annotate" each activity class if one or more of its temporal aspects are directly measured in the system. For example, if the system logs the end time of an activity *Processing*, then the modeler can add the following axiom to the ontology: $Processing \subseteq EndMeasured$, also implicitely stating that $Processing \subseteq EndMeasured \subseteq EndObserved$. Such annotations will serve as a priori knowledge to the reasoner later. Moreover, additional instrumentation knowledge can be encoded in the ontology if modelers subsume the $*Measured$ classes (e.g., name of the logging component, format, reference to source code, etc.).

The general classes for observable data (BO, DO, and EO) provide an abstraction layer that hides the exact source of observability. Inference rules will reference this abstract level to handle and propagate explicit and inferred observability uniformly (Sec. 4.5).

The BI, DI, and EI classes are the extension points of the observability ontology, i.e., the superclasses for implementing observability inference rules, as detailed next.

4.5 Observability inference

Given a partially observable activity model, an OWL-DL reasoner can infer further observable temporal aspects based on rules utilizing Allen interval and structural relations. The ODK provides general inference rule ontologies that encode how explicit measurements (i.e., observability) can be propagated along activity relations.

Users of the ODK can build their custom activity ontology of the SUT (containing the hierarchy of activities and their relations), including the explicit observability of activities (i.e., which activities are directly measured through sensor

instrumentation). Then, inputting the inference rule and user ontologies to a reasoner (e.g., in the form of a top-level ontology that imports the previous ontologies) reveals how certain activity temporal aspects (time of beginning, duration, and/or end) can be calculated through related activities. For example, the reasoning process could infer the following knowledge: the *end time* of activity *Processing* can be inferred from the *end time* of its finishing subactivity $Subactivity_4$ (based on $Rule_i$).

The observability inference is implemented using the class equivalence construct of OWL2. The rules are modeled as OWL classes (e.g., $RuleX \subseteq EndInferred$) with corresponding equivalence axioms as *criteria* (describing an anonymous class in OWL in the form of $RuleX \equiv criteria$). The axioms of criteria usually encode some kind of temporal data propagation rule among activities, while referencing the abstract observability of the involved activities.

When a reasoner infers that an activity type *Processing* satisfies the criteria (i.e., subsumes the corresponding anonymous class), *Processing* becomes part of the class hierarchy of the corresponding $*Observed$ class. E.g., if we have a rule ($RuleX$) about inferring the end time of an activity based on some *criteria* (i.e., $RuleX \equiv criteria$), then the following axiom will be added the ontology if *Processing* "matches" *criteria*:

$$Processing \subseteq criteria \equiv RuleX \subseteq EndInferred \subseteq EndObserved$$

Accordingly, *Processing* will be categorized as an *EndObserved* class, allowing the propagation of the newly inferred knowledge through other rules, continuing until no new knowledge can be inferred.

4.6 Inference rules

The inference mechanism is demonstrated through the simple constraint between the beginning time, duration, and end time of any activity instance: $a^b + a^d = a^e, \forall a \in A \subseteq ACT$. This constraint is the basis of the three core inference rules (Eqs. 1–3) provided by the ODK: if two of the temporal aspects are observable, then the third is inferrable. Rules are encoded through equivalent class axioms and subsume the proper BI, DI or EI inference extension points.

$$A \subseteq (DO \cap EO) \equiv Rule1 \Longrightarrow A \subseteq Rule1 \subseteq BI \subseteq BO \tag{1}$$

$$A \subseteq (BO \cap EO) \equiv Rule2 \Longrightarrow A \subseteq Rule2 \subseteq DI \subseteq DO \tag{2}$$

$$A \subseteq (BO \cap DO) \equiv Rule3 \Longrightarrow A \subseteq Rule3 \subseteq EI \subseteq EO \tag{3}$$

The ODK contains numerous additional inference rules based on Allen interval and structural relations. The following conjunctions of criteria (in the form of

$A \subseteq (criteria_1 \cup ... \cup criteria_n) \implies A \subseteq BO/DO/EO)$ succinctly encode the additional rules for inferring beginnings, durations, and ends, respectively:

$$A \subseteq ((FA \cap \forall hasSubactivity.BO) \cup (\exists starts.BO) \cup (\exists startedBy.BO) \cup (\exists metBy.EO) \cup (\exists hasParentActivity.(BO \cap FA))) \implies A \subseteq BO \tag{4}$$

$$A \subseteq ((\exists hasParentActivity.(TA \cap DO) \cap \forall hasSiblingActivity.DO) \cup (TA \cap \forall hasSubactivity.DO)) \implies A \subseteq DO \tag{5}$$

$$A \subseteq ((\exists meets.BO) \cup (\exists finishes.EO) \cup (\exists finishedBy.EO) \cup (FA \cap \forall hasSubactivity.EO) \implies A \subseteq EO \tag{6}$$

The ODK allows the declaration of additional rule classes by simply subsuming the appropriate $*Inferred$ classes.

Note that an activity A can match multiple rules. For example, if two temporal aspects are observable, A matches one of Eqs. 1–3. However, now all three aspects are observable, so A matches all three rules. In general, the matching inference rules between temporal data define a data flow network, facilitating various data analysis tasks, as detailed in Sec. 6.

4.7 ODK extendability

The ODK operates with high-level and abstract concepts in order to allow extendability with additional concepts, increasing the flexibility and usability of the model in subsequent analysis tasks. Fig. 8 contextualizes the different ODK capabilities in the typical MDE worklfow.

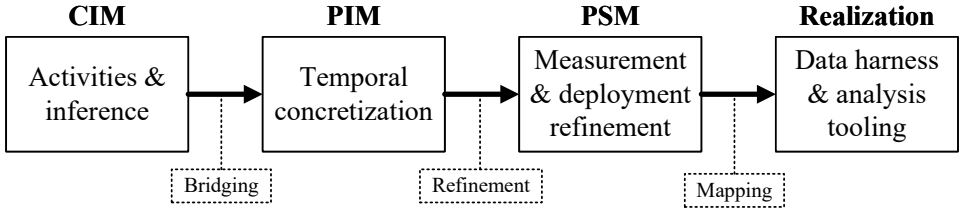


Figure 8: Envisioned MDE flow of activity modelling

The core activity concepts and inference rules comprise a computation-independent model (CIM) for describing observability in a temporal representation-agnostic way. The models at this level (e.g., in Sec. 5) only state knowledge like the beginning of a *Processing* activity is measured. Information about *how* that measurement is acquired, and in *what format*, is omitted. Moreover, inference rules define only the data dependency of calculated measurements – again, omitting the exact computational steps.

The first step towards an actual realization of the analysis process is to enrich the core model with temporal data and corresponding relations. E.g., the OWL

Time ontology[11] defines *Interval*s as "a temporal entity with an extent or duration." Furthermore, the ontology defines the *hasBeginning* relation (among others) between temporal entities (such as intervals) and arbitrary time instants. A simple *bridging* between the two ontologies (e.g., $Activity \equiv Interval$) enriches the activities with actual temporal data formats. The associated temporal data concretize the manner of measurement calculations, but still neglects the exact source and harness of measurement data, thus acting as a platform-independent model (PIM).

Two ODK aspects support the refinement of PIMs into platform-specific models (PSM). On one hand, additional ontologies can refine classes like *BeginningMeasured* to inlcude the source of the measurement data. For example, an extending ontology could define a *BeginningLogged* subclass of *BeginningMeasured*, providing details about the log format, source software component, and the semantic structure of the message, all aiding the extraction of measurement information in a log processing pipeline.

On the other hand, the ODK provides an *executedBy* relation to associate an *Activity* type with a *Service* type (e.g., *Endorsement* activities are *executedBy PeerService*s). Extending ontologies can build on this relation to further model the deployment information related to a certain environment where the SUT is operated. For example, a deployment ontology could maintain information about a HLF network, where each service instance is located on a certain Kubernetes[12] node, in a cluster comprised of several virtual machines, hosted on specific hardware components.

Finally, a technology stack realizing the actual data analysis flow can utilize all levels of the final, rich model to uncover the root cause of an anomalous activity duration/latency (partially demonstrated in Sec. 6.4), even if stemming from the lowest level of the infrastructure.

5 Case study: Modeling TPC-C on Fabric

Performance benchmarks serve as platform-agnostic workload specifications representative for a given domain, facilitating the comparison of different backend platform implementations under reproducible conditions. The benchmark plays the role of a platform-independent model (PIM) in MDE terminology, while the emulated clients and database engine make it platform-specific. The section introduces a compositional model of the TPC-C workload executed on HLF, using the presented ODK concepts as case study.

5.1 Modeling the TPC-C benchmark

TPC-C is a mature online transaction processing (OLTP) benchmark inspired by the typical activities of a wholesale supplier. TPC-C uses a mix of five transaction

[11] https://www.w3.org/TR/owl-time/
[12] https://kubernetes.io/

types – with varying complexity – to be executed against a rich database schema (HLF in the case study).

The execution of a TPC-C transaction by an emulated client has the following, strictly sequential composition of steps: (1) the client selects a transaction type (*Menu selection*); (2) then fills the required inputs for the request (*Fill inputs*); (3) then the database engine executes the transaction (*Execute TX*); (4) and finally the client takes some time to think about the next transaction (*Think time*) before starting the next cycle. The model of this client cycle plays the role of the workload ontology in Fig. 5.

Accordingly, the activity model (Fig. 9) declares a top-level/root sequential activity, having four subactivities. The *Menu selection*, *Fill inputs*, and *Think time* subactivities are atomic activities that simply emulate user behavior through artificial delays with specified distributions.

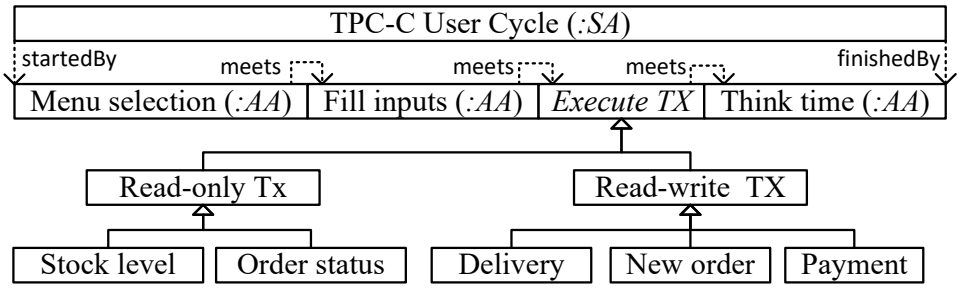


Figure 9: TPC-C transaction execution scheduling

The exact composition of the *Execute TX* activity is specific to the database engine, thus its type does not subsume any of the *Activity* ontology classes. The exact type binding is the task of a platform-workload bridging ontology that maps the platform request execution activities to the *Execute TX* activity.

The TPC-C transaction types are further categorized based on whether they are read-only, or read-write requests, making the bridging easier to platforms that differentiate between the execution of the two categories (like HLF does).

5.2 Modeling the Hyperledger Fabric consensus

During the benchmark measurement, a HLF network served as the "database engine." The novelty of HLF is its execute-order-validate style consensus mechanism, breaking with the traditional order first approaches [5]. However, its performance characterization is still incomplete. The case study models the detailed HLF consensus mechanism, enriched with client-side observations provided by the Hyperledger Caliper workload generator.

The concepts and consensus steps of HLF are detailed in [5] or in the official documentation.[13] The section focuses only on the composition of activities (and

[13] https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

not on their technical descriptions) to demonstrate that deep domain knowledge is not required during the guided performance analysis tasks. Note that creating the model, however, requires familiarity with the modeled platform, but ideally it is the responsibility of the designers or platform experts to create such a model.

Fig. 10 details the high-level, sequential steps of the HLF transaction life-cycle. The model plays the role of the platform ontology in Fig. 5.

Clients first assemble and send a transaction proposal to one or multiple peers for *parallel* simulation/endorsement and wait for *all* results (*Awaiting Endorsement* activity) to arrive, modeled by an associated *WaitForAll* synchronization semantic. Once the results are available, the client then sends them to the ordering service and waits for a notification from the network that the transaction was successfully committed or not (*Awaiting Ordering and Validation* activity).

The ordering and validation phase is modelled by two consecutive subactivities: *Block inclusion* and the client *Awaiting Validation* from *any* peer (denoted by a *WaitForAny* synchronization semantic). It is important to note, that the *Awaiting Validation* activity is not a dedicated, explicitly observable activity of the client. It is artificially introduced for convenience to separate the ordering and the validation steps for detailed analysis. This choice demonstrates that the activity model is constructed in a way to facilitate detailed performance analysis, rather than be a technically faithful representation of the platform.

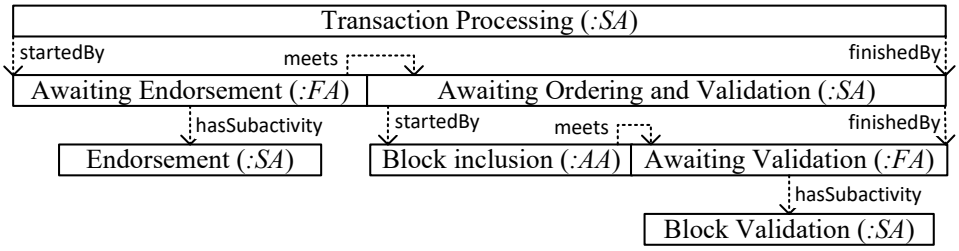


Figure 10: High-level steps of the HLF consensus

The endorsement activity (Fig. 11) consists of the peer receiving the proposal, calling the required chaincode, then returning the result to the client. On the platform level, the *Chaincode Call* activity type is not specified to enable refinement by different use cases, detailed in the next section.

The block validation and commit process of peers is modelled by a hierarchy of activity sequences (Fig. 12). The validation begins by the ordering service delivering the new block to the peer (*Getting Block*). Then the peer checks the block payload and fetches any private data (a privacy feature of HLF) required for further validation (*Check Payload* and *Fetch pvt. data* activities).

The *State validation and commit* step is refined into further subactivities. First, the state modifications of transactions are validated (*State validation*). Then the raw block content is committed to the blockchain storage (*Block Commit*). Next the state modification of valid transactions are committed to the world state database (*State Commit*). Finally, the history database is updated with the data accesses of

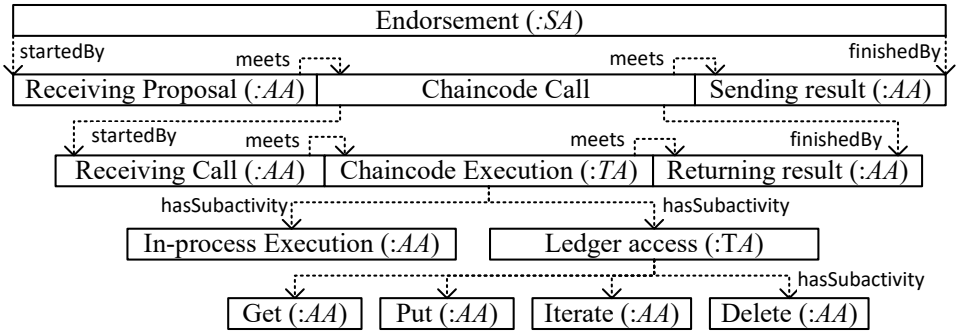


Figure 11: Steps of the endorsement activity

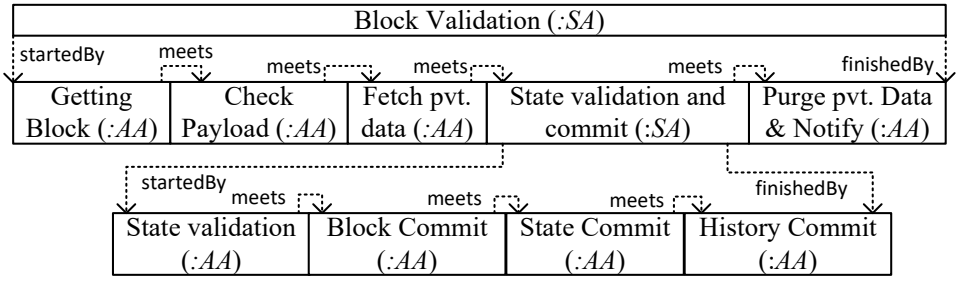


Figure 12: Steps of the validation and commit activity

committed transactions (*History Commit*).

Finally, the peer purges stale private data and sends a notification about the block commit to subscribed clients. Once a client receives a notification about a block/transaction, the transaction life-cycle is considered complete.

5.3 TPC-C and HLF bridge ontology

The case study contains a final ontology that maps/bridges the TPC-C and HLF concepts, achieving the *TPC-C on HLF* model. The mapping plays the role of the platform-workload bridge ontology in Fig. 5.

On one hand, the TPC-C case study chaincode was instrumented to measure the raw execution time of the chaincode. This allows the observation of peer-chaincode communication activities and differentiate between in-process execution and ledger access times (lower part of Fig. 11). The exact control flow of the chaincode is not modelled, alternating activities are used instead to focus only on the duration of subactivities, and not on their order.

On the other hand, the bridge also refines the *Execute TX* class of the TPC-C ontology. Due to the *Read-only TX* and *Read-write TX* class hierarchy, the following equivalence axioms are enough to specify that the workload is executed on HLF: i) *Read-write TX* $\equiv$ *Transaction Processing* and ii) *Read-only TX* $\equiv$ *Query*

Processing (which is a simplified version of transaction processing, containing only the endorsement activity hierarchy, without further ordering or validation).

6 Systematic measurement data analysis

At this point, the workload and platform ontologies are combined, and the measured activities are annotated with the appropriate *BM/DM/EM* observability classes. Inputting the ODK and user-defined ontologies to an OWL-DL reasoner will propagate the measured activity aspects throughout the rest of the model by flagging activities with different inference rule classes. The added classes denote how the beginning, duration and end of a flagged activity can be calculated based on other activity observations.

The case study imported the bridged TPC-C and Fabric (Sec. 5.3), and inference ontologies (Sec. 4.5) into a top-level ontology (as depicted by Fig. 5). Then, the Hermit OWL-DL reasoner [35] (present as a built-in reasoner plugin in the Protégé ontology tool [36]) performed a subsumption relation inference. The resulting (output) ontology contained additional knowledge about how to derive certain temporal aspects of TPC-C and Fabric activities (even when directly unobserved by sensors) based on the explicit measurement data.

The added rule classes define *relations* between the temporal data of different activities. The following subsections provide examples for how such relations can be exploited to:

1. correlate and validate the distributed measurement data;
2. derive further, directly not measured (i.e., indirect) temporal data;
3. validate the conformance of measurement data to the activity model;
4. and systematically guide the bottleneck analysis tasks.

6.1 Correlate and validate measurement data

Online services today exhibit a shift towards micro-service architectures to facilitate different DevOps tasks (e.g., rapid continuous delivery and deployment) and increase certain extra-functional properties of systems (e.g., availability, maintainability, fault tolerance, scalability). Accordingly, an end user request will traverse many services and corresponding components during processing. The same phenomenon is inherently present in distributed, peer-to-peer systems, such as HLF.

In most cases a unique correlation/trace identifier is associated with each request to facilitate its tracing across component boundaries. HLF, for example, associates a unique transaction identifier (TX ID) with each client request, calculated from the client's identity and the time the transaction was constructed. When network components provide logs about certain transaction steps, they also log the corresponding TX ID along with the trace data.

A prerequisite of reconstructing a detailed activity timeline of transactions is the collection and correlation of such distributed traces. Novel observability frameworks, such as OpenTelemetry,[14] may provide means to collect traces across component boundaries. For example, services utilizing OpenTelemetry can also send the collected traces (as metadata) along with the requests to other system components. Even though such approaches can centralize trace collection to a certain level, it is a rather intrusive instrumentation choice, hindering adoption by existing systems (such as HLF[15]).

Instead, many systems opt to provide request trace data utilizing their already existing logging capabilities. In this case, distributed transaction traces must be collected and correlated using a separate monitoring stack, which presents its own challenges (but at least it is separate from the core system functionality). Having a detailed activity model for distributed transaction processing (such as the HLF consensus process) can facilitate the correlation and availability check of traces.

A HLF network setup usually contains the following trace sources (Fig. 13):

- optional end-to-end traces logged by the client (Caliper, in this case), with an associated TX ID;
- optional traces logged by chaincodes (one for each executing peer), with an associated TX ID;
- chaincode call traces logged by the peer nodes (one for each executing peer), with an associated *shortened* TX ID (first 8 characters only);
- block validation and commit traces logged by the peer nodes (one for each peer), with an associated block ID;
- and block creation traces logged by the leader orderer node, with an associated block ID.

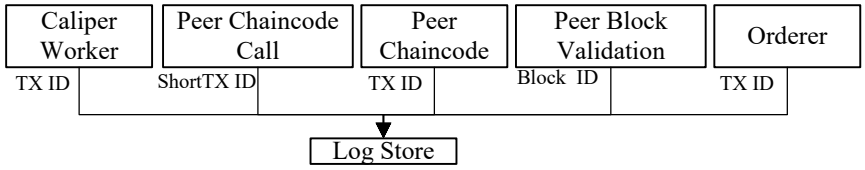


Figure 13: Different trace sources of a transaction

The activity model of the case study defined the measured temporal data of activities, associated with the service types logging them. Accordingly, the prerequisite trace correlation step simply followed the structure of the model to check whether all supposedly measured data are available from all sources.

[14] https://opentelemetry.io/
[15] https://hyperledger.github.io/fabric-rfcs/text/0000-opentelemetry-tracing.html

The check revealed two anomalies:

1. there were 4334 transactions with missing traces;
2. there were 2 transactions with more traces than required.

Case 1 had an interesting symmetry in it: there were 2167 transactions where *all* Caliper-side traces were missing; and there were 2167 transactions where *all* other (non-Caliper) traces were missing. This lead to the hypothesis that one half is actually corresponding to the other half.

Since the non-Caliper traces constituted a complete data set on their own, the focus of investigation was Caliper's Fabric integration. Further transaction metadata analysis revealed that all mismatched traces were HLF queries. Finally, the investigation revealed a *bug in Caliper's query submitting logic*.[16] Caliper created a TX ID for the request, but did not pass it along to the HLF SDK, which in turn created a new (and different) TX ID, unknown to Caliper. This resulted in client-side traces having a different TX ID than HLF-side traces.

Case 2 was a similarly peculiar anomaly. Two transactions had chaincode call traces from peers that did not even execute those transactions. Closer inspection revealed that the shortened TX IDs contained a duplicate, i.e., two different TX IDs had the same shortened (8 characters) versions. Accordingly, the pairing of traces was not unique, two transaction got each others chaincode call traces.

Luckily, the correct traces could be restored without data loss through temporal correlation: the conflicting transactions were executed well apart in time. However, if all peers would have executed those transactions, then the short TX ID conflict would have gone unnoticed until later in the analysis workflow (Sec. 6.3). The anomaly showed that reducing the information carried by trace correlation IDs is highly discouraged.

6.2 Deriving indirect measurement data

The final activity model of the HLF consensus refines a transaction into 28 hierarchical steps even if only a single peer endorses and validates transactions. In general, the number of activities corresponding to a transaction is $5+13*E+10*V$, where $E \in \mathbb{N}^+$ is the number of endorsing peers for a transaction, and $V \in \mathbb{N}^+$ is the total number of peers in the network (since every peer validates transactions).

Moreover, each activity has three associated temporal data: its beginning time, duration, and end time. Accordingly, the volume of temporal can quickly increase with the network size and the number of analysed transactions. For the sake of readability, let us assume that only a single peer endorses and validates transactions, resulting in 84 potentially observable temporal data for the 28 activities of *each* transaction.

Figs. 14–16 depict each activity and their corresponding temporal data (beginning, duration, and end). Black-filled shapes mark the directly measured data

[16] https://github.com/hyperledger/caliper/issues/1187

points. Using a component-off-the-shelf (COTS) HLF as SUT and Caliper as workload generator, there are 18 directly measured data points:

- Caliper marks: the beginning of a transaction; the end time when all endorsements arrive; and the end time when a notification is received about a committed block/transaction.
- Orderer nodes mark the end time when a new block is created.
- Peer nodes mark: the beginning, duration and end of a chaincode call; the end time when a block is received from an orderer; the end time and duration for checking the payload of a new block; and the end time and duration (including durations of some substeps) for validating and committing a block.
- The TPC-C chaincode implementation marks the start time, duration, and end time of the actual chaincode program execution.

The arrows in Figs. 14–16 symbolize the direction of measurement data propagation, i.e., $A \longrightarrow B$ means that data B is calculated from data A (and possibly from other data in cases like $A \longrightarrow B \longleftarrow C$). The arrows essentially represent inference rules in the model, e.g., stating that the start time of an activity can be calculated from its end time and duration (like in the case of the *State validation and commit* activity).

As shown in the figures, the directly measured temporal aspects are sufficient to completely observe the entire activity hierarchy through measurement propagation. If that were not the case, then the missing data propagation paths would identify the places where the SUT needs additional sensor instrumentation to allow for more detailed observability.

Note that Figs. 14–16 are just a single, simplified view of a more complex data flow network determined by the applicable inference rules. The rigorous formal analysis of such data flow networks (in the context of temporal data) is subject to future work.

Moreover, the example assumes a single-peer HLF network. If the network consists of more than one peer, then the single-peer assumption is achieved by reducing the replicated endorsement and validation activities to a single instance by disregarding the non-bottleneck instances:

1. Since transaction endorsements have a $WaitForAll$ synchronization semantic, keep only the longest running (i.e., the slowest) *Endrosement* activity and its subactivities.
2. Since block validations have a $WaitForAny$ synchronization semantic, keep only the shortest running (i.e., the fastest) $BlockValidation$ activity and its subactivities.

At this point, a data analyst can use the formal data flow network to systematically derive new temporal data about the SUT's activities. However, an additional validation step is still needed to ensure not only the cleanness and richness, but also the correctness of the measurement data (or the model).

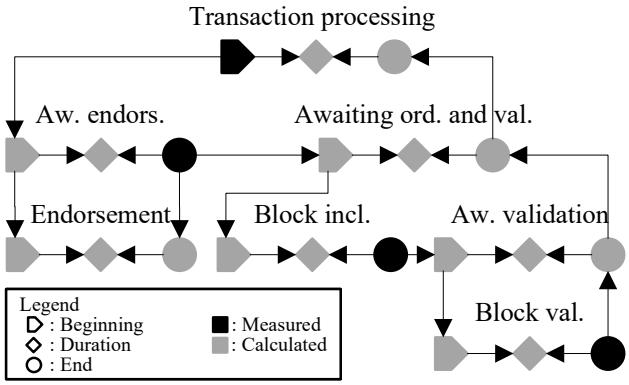


Figure 14: Measurement propagation for high-level HLF activities

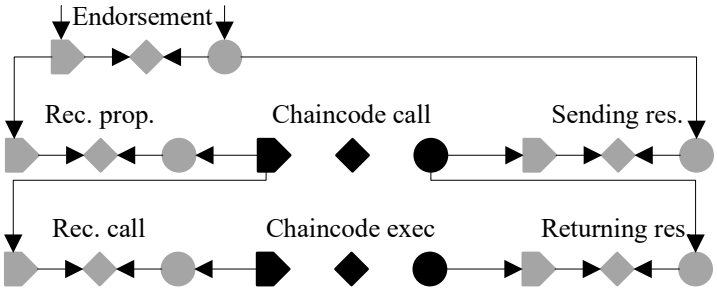


Figure 15: Measurement propagation for the endorsement-related activities

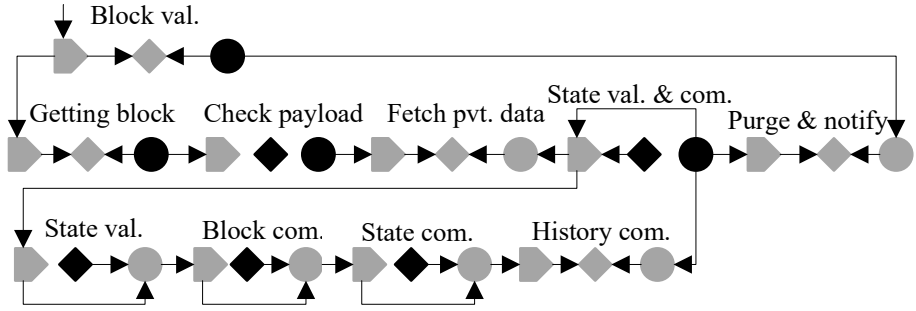


Figure 16: Measurement propagation for the correct validation-related activities

6.3 Model and measurement data validation

Validating the measurement data is an important step to ensure the correctness of data analysis findings and insights. The proposed model-guided approach necessitates the following validation steps before proceeding to the performance analysis tasks:

1. checking the conformance of measurement data to the activity model;
2. and checking the consistency of the measurement data itself.

6.3.1 Detecting modelling errors

The following scenario demonstrates how model conformance checks can reveal activity modelling errors. Such errors can be common if the model is reverse-engineered by others than the platform developers (like in this case study).

For example, HLF peers log the *State validation and commit* activity details using the following message format: `[mychannel] Committed block ... in 26ms (state_validation=3ms block_and_pvtdata_commit=16ms state_commit=3ms)`.

Accordingly, a previous version of the consensus model refined the *State validation and commit* activity as having only three subactivities (state validation, block commit, and state commit, as indicated by the log format). Fig. 17 shows the temporal data propagation for the initial version.

Note how the (directly unobserved) beginning time of the *State validation* subactivity can be calculated in two different ways (highlighted arrows in Fig. 17): i) based on sibling activity data; ii) and/or directly from parent activity data. There should not be any difference between the two paths in the case of a correct model and instrumentation. Validating this assumption requires checking whether the beginning times of the *State validation* subactivities coincide with the beginning times of the *State validation and commit* parent activities for every transaction, as required by the *startedBySubactivity* relation among the two activity classes.

However, performing the check revealed that the equality constraint was violated for *every* transaction. The *State validation* activities always started later than their parent activities, indicating the presence of a hidden subactivity. Moreover, the magnitude of the missing time was sometimes non-negligible (Fig. 18), i.e., it could not be considered a measurement noise, thus warranting further investigation.

As it turns out, the format of the log message was misleading and not all relevant subactivities were listed in the message. The source code inspection of HLF revealed that there is an other non-negligible subactivity performed during *State validation*

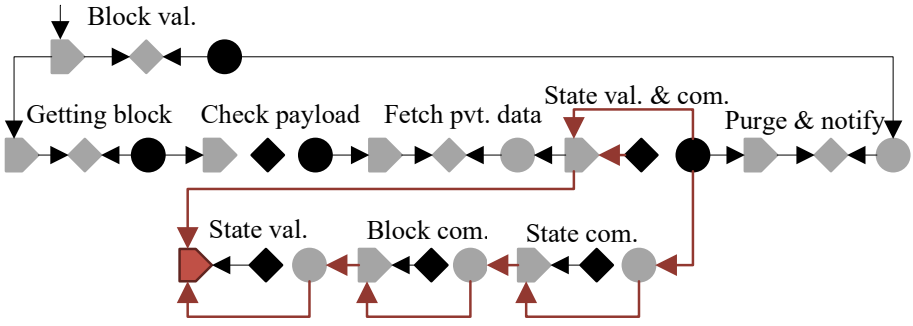


Figure 17: Measurement propagation for the initial validation-related activities

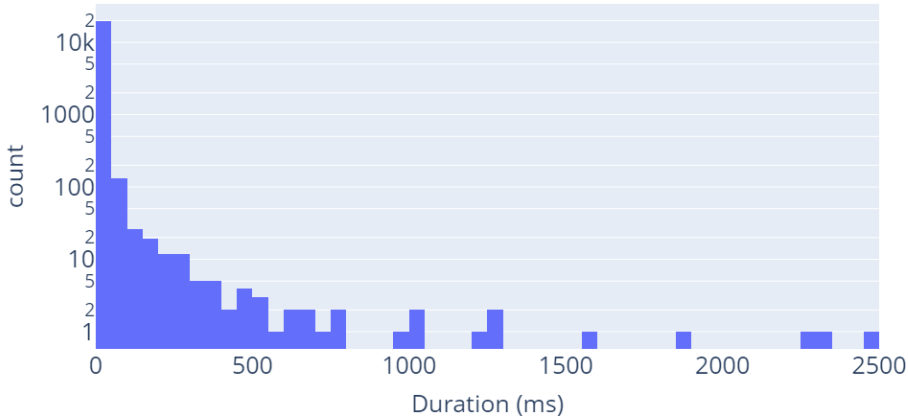


Figure 18: Frequency distribution of missing validation time durations

and commit, namely committing the state modifications of a transactions to a history database. Accordingly, the final model of the HLF consensus was extended with the *Commit history* subactivity (Fig. 16).

Note that measurement noises are a common occurrence in complex, especially high-throughput or overloaded systems. The measurement data conformance check also revealed some inconsistencies around the *Check payload* activity. Calculating the beginning time of the activity from its own end time and duration yielded a different result than propagating the end time of its immediate predecessor *Getting block* activity. Even though the propagation path is short and simple, it still violated the modeled activity relationship. However, the magnitude of the amount of unaccounted time for the activity (Fig. 19) is negligible.

One probable explanation could be that the missing time is a side-effect of the logging mechanism: the measured duration was calculated based on times $start_{calc}$ and end_{calc}, while the logging library marked the log message with an $end_{log} > end_{calc}$ timestamp, and end_{log} was taken as the measured end time by the log processing pipeline. An other explanation could be that negligible activities were performed between the two modeled activities that can be safely ignored during performance analysis.

6.3.2 Detecting measurement errors

The systematic data propagation can also aid the detection of measurement (or measurement setup) errors. The missing subactivity issue manifested itself as unaccounted time in the transaction timeline. The other important symptom of inconsistent measurement data is negative durations.

The analysis showed negative *Receiving proposal* activity durations upon measurement data validation. The duration in question is a derived measurement. Its value is indirectly calculated as the difference between the beginning time of calling a chaincode (*Chaincode call* activity) and the beginning time of creating

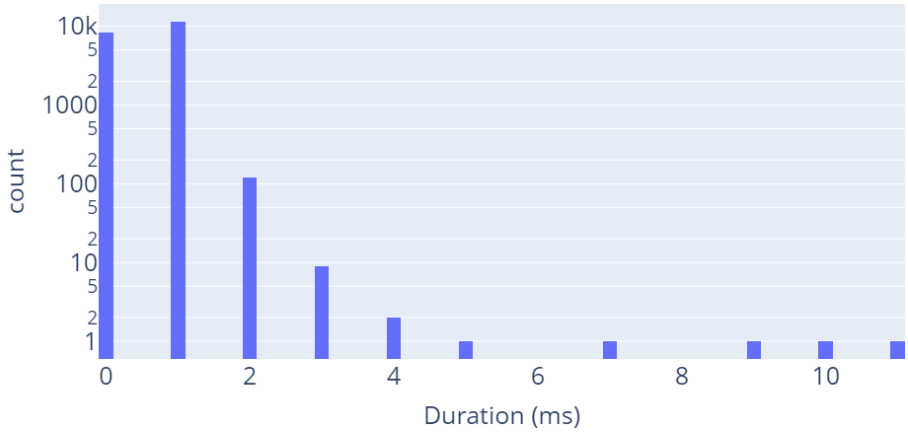


Figure 19: Frequency distribution of missing payload check time durations

a transaction (*Transaction processing* activity), both data being direct measurements. A negative duration result would mean that the chaincode is called before the transaction is even constructed, which is a serious *event causality violation*.

Note that the two direct measurements (the bases of the duration calculation) originate from two different (physical) components in the distributed network: the beginning of *Transaction processing* is captured by Hyperledger Caliper (i.e., the HLF client), while the beginning of the Chaincode call is logged by the HLF peer nodes. Fig. 20 shows the *Receiving Proposal* durations for each transaction over the time of the SUT measurement and reveals a curious trend: the anomalous durations smoothly oscillate around zero over time, i.e., negative durations are not that isolated and sporadic. Moreover, Fig. 20 depicts the activity data of each transaction after non-bottleneck endorsement activities have been eliminated, as outlined in Sec. 6.2. Correspondingly, different *Receiving Proposal* activity durations may originate from different peer nodes of the network.

Combining the observations with the outlined assumptions results in the following working hypothesis: the system clock of a peer node periodically drifts out of sync from the other components. Measurement setup investigations later revealed that network nodes used a default, light-weigh time synchronization service instead of a more sophisticated one that provides higher precision.

Measurement errors of such a low magnitude was deemed negligible in the previous section (Fig. 19). However, in this case, the presence of event causality violations shadows the usually insignificant magnitude of the actual measurement error. For example, process mining approaches can produce significantly different results in the presence of such causality violations.

Considering only the atomic activities of the HLF consensus model results in the low-level sequence of steps of the transaction life-cycle. Inputting the measurement data of such activities into a process mining algorithm should result in the process of Fig.21, assuming that the measurement data reflects the correct causality of

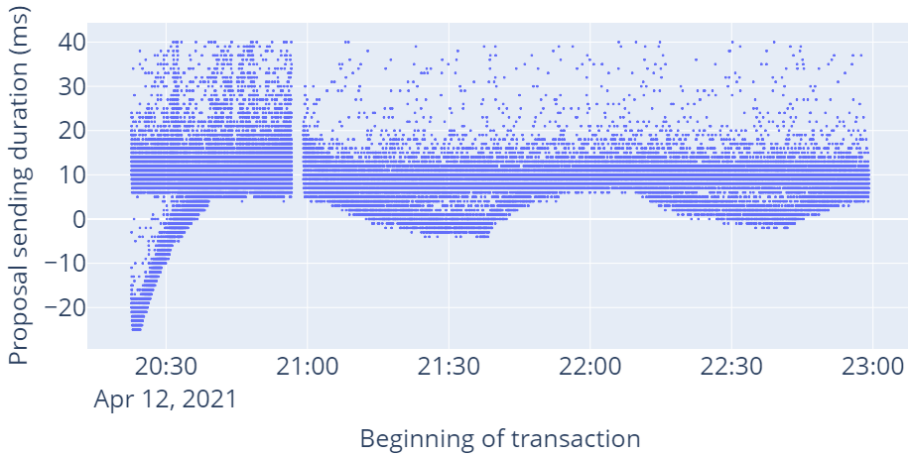


Figure 20: Effect of misaligned system clocks over time

events. However, the presence of causality violations in the input temporal data can lead to an incorrect process model (Fig. 22). Such models can hinder the correct understanding and insights of the SUT (that would be the goal of process mining) even for experienced HLF domain experts.

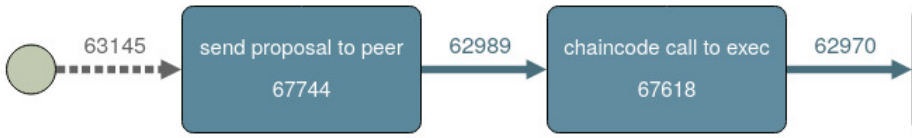


Figure 21: Process mining result without causality violations

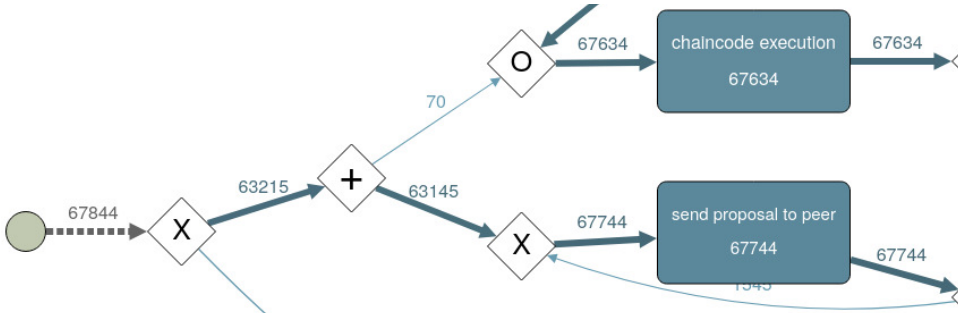


Figure 22: Process mining result with causality violations

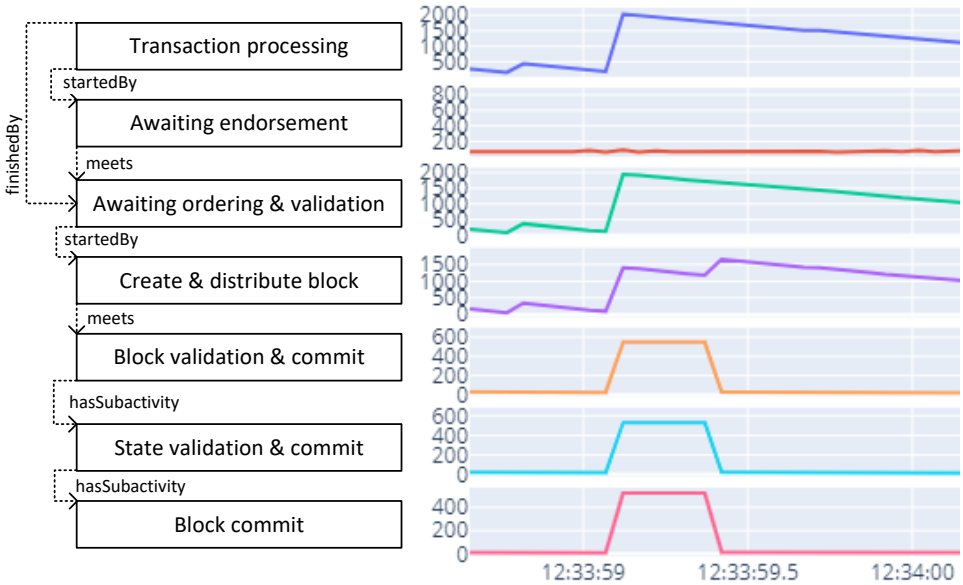


Figure 23: A partial HLF activity time series hierarchy demonstrating a latency anomaly.

Correspondingly, systematically cleaned and validated data is a must if the data analysis workflow incorporates formal approaches. The proposed approach and supporting ontology models enable rigorous (and possibly automated) measurement data validation before performing further performance analysis tasks.

6.4 Guided bottleneck identification

The primary goal and advantage of the proposed approach is that by the time the data analysts reach the actual performance analysis task, the available measurement data is validated, cleaned, and structured among semantically precise relations. The last section demonstrates how bottleneck identification and the root cause analysis of latency anomalies become intuitive and easy-to-perform tasks, given the proper input data.

Let us assume that an end-to-end latency spike is detected on the client-side, classified as an anomaly (the exact anomaly detection methods are out of the scope of this paper). Fig. 23 demonstrates how the hierarchical activity data aids the root cause analysis of the anomaly, uncovering bottlenecks contributing to the latency spike.

The analysis employs a drill-down approach using the parent-subactivity hierarchy relations to gradually pinpoint significant latency contributors. At first, the latency of the high-level transaction processing subactivties are considered. Since endorsement times seem constant during the anomaly, the endorsement activity is

dismissed as bottleneck and root cause. The ordering and validation subactivity, however, exhibits the same latency trend as the end-to-end anomaly. Correspondingly, it becomes the next activity of interest.

At this point, the subactivity latency trends show an interesting pattern. Neither the block creation, nor the block validation subactivites show the same trend as their anomalous parent activity. However, both indicate deviation from their previous baseline latency characteristics. Accordingly, the hierarchical exploratory process supports the identification of *multivariate root causes*.

Block creation is a leaf activity element in the HLF consensus model, thus further root cause analysis along this path would require additional instrumentation or the detailed inspection of corresponding computing resource utilizations. The other prominent root cause path is the block validation activity. Further drill-down steps reveal that the atomic block commit activity caused the latency spike in this path. However, it must be noted that while the block commit anomaly is a short transient spike, the block creation latency needs more time to settle, hinting at some system statefullness and *memory in the performance domain* (probably resulting from a queuing mechanism).

Nevertheless, the hierarchical and systematic approach allows the intuitive and quick identification of bottleneck activities of the SUT. Given the activities of interest, the next analysis steps include the correlation of bottleneck activity latencies with the corresponding component *resource utilizations*, or with the *characteristics of the workload*. Such correlations can answer the question whether the anomaly is caused by resource limits, or it is not really an anomaly, but a change in the presumed workload affected the exprected performance characteristics of the SUT. Such analysis, however, is outside the scope of this paper.

7 Conclusion

The increasing volume and dimensionality of performance measurement data necessitate the rigorous model-based support of data analysis tasks, such as bottleneck identification. While traditional DevOps approaches already benefit from MDE, performance data analysis lacks such support.

The paper proposed an ontology-guided workflow (and presented the corresponding ODK) for modeling the composition of complex platform activities and their explicit observability. The ODK also supplies numerous inference rules to reason about the implicit observability of activities, creating a rich model serving as a strong formal basis for later performance analysis tasks.

A representative case study demonstrated the advantages of the approach: a model-guided drill-down bottleneck identification process for a TPC-C benchmark workload executed on a HLF network. The current work aims at the integration of domain-specific knowledge in performance analysis into a core ontology, providing a strong formal foundation for measurement data analysis and performance monitoring of complex systems.

References

[1] Abelló, Alberto, Romero, Oscar, Pedersen, Torben Bach, Berlanga, Rafael, Nebot, Victoria, Aramburu, María José, and Simitsis, Alkis. Using semantic web technologies for exploratory OLAP: A survey. *IEEE Trans. on Knowledge and Data Engineering*, 27(2):571–588, 2015. DOI: 10.1109/TKDE.2014.2330822.

[2] Allen, James F. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983. DOI: 10.1145/182.358434.

[3] Allen, James F. and Ferguson, George. Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531–579, 1994. DOI: 10.1007/978-0-585-28322-7_7.

[4] Androulaki, Elli, De Caro, Angelo, Neugschwandtner, Matthias, and Sorniotti, Alessandro. Endorsement in Hyperledger Fabric. In *Proceedings — 2nd IEEE Int. Conf. on Blockchain*, pages 510–519. Institute of Electrical and Electronics Engineers Inc., 2019. DOI: 10.1109/Blockchain.2019.00077.

[5] Androulaki, Elli et al. Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proc. of the Thirteenth EuroSys Conf.*, 2018. DOI: 10.1145/3190508.3190538.

[6] Ardagna, Claudio A., Bellandi, Valerio, Ceravolo, Paolo, Damiani, Ernesto, Bezzi, Michele, and Hebert, Cedric. A Model-Driven Methodology for Big Data Analytics-as-a-Service. In *Proc. of the IEEE 6th Int. Congress on Big Data*, pages 105–112, 2017. DOI: 10.1109/BigDataCongress.2017.23.

[7] Baliga, Arati, Solanki, Nitesh, Verekar, Shubham, Pednekar, Amol, Kamat, Pandurang, and Chatterjee, Siddhartha. Performance characterization of Hyperledger Fabric. In *Crypto Valley Conf. on Blockchain Technology*, pages 65–74, 2018. DOI: 10.1109/CVCBT.2018.00013.

[8] Becker, Steffen, Koziolek, Heiko, and Reussner, Ralf. The Palladio component model for model-driven performance prediction. *J. of Systems and Software*, 82(1):3–22, 2009. DOI: 10.1016/j.jss.2008.03.066.

[9] Bellini, Pierfrancesco, Mattolini, Riccardo, and Nesi, Paolo. Temporal logics for real-time system specification. *ACM Computing Surveys (CSUR)*, 32(1):12–42, 2000. DOI: 10.1145/349194.349197.

[10] Bergman, Sara, Asplund, Mikael, and Nadjm-Tehrani, Simin. Permissioned blockchains and distributed databases: A performance study. *Concurrency and Computation: Practice and Experience*, 32(12):e5227, 2020. DOI: 10.1002/cpe.5227.

[11] Cerveira, Frederico, Kocsis, Imre, Barbosa, Raul, Madeira, Henrique, and Pataricza, András. Exploratory data analysis of fault injection campaigns. In *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, pages 191–202, 2018. DOI: 10.1109/QRS.2018.00033.

[12] Chen, Liming, Nugent, Chris D., and Wang, Hui. A knowledge-driven approach to activity recognition in smart homes. *IEEE Trans. on Knowledge and Data Engineering*, 24(6):961–974, 2012. DOI: 10.1109/TKDE.2011.51.

[13] Foschini, Luca, Gavagna, Andrea, Martuscelli, Giuseppe, and Montanari, Rebecca. Hyperledger Fabric blockchain: Chaincode performance analysis. In *IEEE Int. Conf. on Communications*, 2020. DOI: 10.1109/ICC40277.2020.9149080.

[14] Garg, Supriya, Nam, Julia Eunju, Ramakrishnan, I.V., and Mueller, Klaus. Model-driven visual analytics. In *VAST'08 - IEEE Symp. on Visual Analytics Science and Technology, Proc.*, pages 19–26, 2008. DOI: 10.1109/VAST.2008.4677352.

[15] Garlan, D., Cheng, S.W., Huang, A.C., Schmerl, B., and Steenkiste, P. Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer*, 37(10):46–54, 2004. DOI: 10.1109/MC.2004.175.

[16] Gilmore, Stephen and Hillston, Jane. The PEPA workbench: A tool to support a process algebra-based approach to performance modelling. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 353–368. Springer, 1994. DOI: 10.1007/3-540-58021-2_20.

[17] Gorenflo, Christian, Lee, Stephen, Golab, Lukasz, and Keshav, Srinivasan. FastFabric: Scaling Hyperledger Fabric to 20,000 transactions per second. In *IEEE Int. Conf. on Blockchain and Cryptocurrency*, pages 455–463, 2019. DOI: 10.1002/nem.2099.

[18] Gupta, Himanshu, Hans, Sandeep, Aggarwal, Kushagra, Mehta, Sameep, Chatterjee, Bapi, and Jayachandran, Praveen. Efficiently processing temporal queries on Hyperledger Fabric. In *Proc. IEEE 34th Int. Conf. on Data Engineering, ICDE 2018*, pages 1435–1440. Institute of Electrical and Electronics Engineers Inc., 2018. DOI: 10.1109/ICDE.2018.00167.

[19] Hao, Yue, Li, Yi, Dong, Xinghua, Fang, Li, and Chen, Ping. Performance analysis of consensus algorithm in private blockchain. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 280–285. Institute of Electrical and Electronics Engineers Inc., 2018. DOI: 10.1109/IVS.2018.8500557.

[20] Hashem, Ibrahim Abaker Targio, Yaqoob, Ibrar, Anuara, Nor Badrul, Mokhtar, Salimah, Gani, Abdullah, and Khanb, Samee Ullah. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015. DOI: 10.1016/j.is.2014.07.006.

[21] Helaoui, Rim, Niepert, Mathias, and Stuckenschmidt, Heiner. Recognizing interleaved and concurrent activities using qualitative and quantitative temporal relationships. In *Pervasive and Mobile Computing*, volume 7, pages 660–670. Elsevier B.V., 2011. DOI: 10.1016/j.pmcj.2011.08.004.

[22] Hermanns, Holger, Herzog, Ulrich, and Katoen, Joost-Pieter. Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1-2):43–87, 2002. DOI: 10.1016/S0304-3975(00)00305-4.

[23] Horrocks, Ian, Kutz, Oliver, and Sattler, Ulrike. The even more irresistible SROIQ. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'06, page 57–67. AAAI Press, 2006.

[24] Inagaki, Tatsushi, Ueda, Yohei, Nakaike, Takuya, and Ohara, Moriyoshi. Profile-based detection of layered bottlenecks. In *Proc. of the 2019 ACM/SPEC Int. Conf. on Performance Engineering*, pages 197–208. ACM, 2019. DOI: 10.1145/3297663.3310296.

[25] Javaid, Haris, Hu, Chengchen, and Brebner, Gordon. Optimizing validation phase of Hyperledger Fabric. In *IEEE Computer Society's Annual Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, pages 269–275, 2019. DOI: 10.1109/MASCOTS.2019.00038.

[26] Jiang, Lili, Chang, Xiaolin, Liu, Yuhang, Mišić, Jelena, and Mišić, Vojislav B. Performance analysis of Hyperledger Fabric platform: A hierarchical model approach. *Peer-to-Peer Networking and Applications*, 13(3):1014–1025, 2020. DOI: 10.1007/s12083-019-00850-z.

[27] Klein, John, Gorton, Ian, Alhmoud, Laila, Gao, Joel, Gemici, Caglayan, Kapoor, Rajat, Nair, Prasanth, and Saravagi, Varun. Model-driven observability for big data storage. In *13th Working IEEE/IFIP Conf. on Software Architecture, WICSA 2016*, pages 134–139, 2016. DOI: 10.1109/WICSA.2016.27.

[28] Kocsis, Imre, Klenik, Attila, Pataricza, András, Telek, Miklós, Deé, Flórián, and Cseh, Dávid. Systematic performance evaluation using component-in-the-loop approach. *Int. Journal of Cloud Computing*, 7(3-4):336–357, 2018. DOI: 10.1504/IJCC.2018.095401.

[29] Kuzlu, Murat, Pipattanasomporn, Manisa, Gurses, Levent, and Rahman, Saifur. Performance analysis of a Hyperledger Fabric blockchain framework: Throughput, latency and scalability. In *2nd IEEE Int. Conf. on Blockchain*, pages 536–540, 2019. DOI: 10.1109/Blockchain.2019.00003.

[30] Lamport, Leslie. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 16(3):872–923, 1994. DOI: 10.1145/177492.177726.

[31] Lamport, Leslie. *Specifying concurrent systems with TLA+*. In *Calculational System Design*, pages 183–247. IOS Press, Amsterdam, 1999.

[32] Lowe, Gavin. Probabilistic and prioritized models of timed CSP. *Theoretical Computer Science*, 138(2):315–352, 1995. DOI: 10.1016/0304-3975(94)00171-E.

[33] Meditskos, Georgios, Dasiopoulou, Stamatia, Efstathiou, Vasiliki, and Kompatsiaris, Ioannis. Ontology patterns for complex activity modelling. In *Proc. International Workshop on Rules and Rule Markup Languages for the Semantic Web*, volume 8035 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2013. DOI: 10.1007/978-3-642-39617-5_15.

[34] Miyamae, Takeshi, Honda, Takeo, Tamura, Masahisa, and Kawaba, Motoyuki. Performance improvement of the consortium blockchain for financial business applications. *Journal of Digital Banking*, 2(4):369–378, 2018.

[35] Motik, Boris, Shearer, Rob, and Horrocks, Ian. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36(1):165–228, 2009. DOI: https://doi.org/10.1613/jair.2811.

[36] Musen, Mark A. The protégé project: A look back and a look forward. *AI matters*, 1(4):4–12, 2015. DOI: 10.1145/2757001.2757003.

[37] Nakaike, Takuya, Zhang, Qi, Ueda, Yohei, Inagaki, Tatsushi, and Ohara, Moriyoshi. Hyperledger Fabric performance characterization and optimization using GoLevelDB benchmark. In *IEEE Int. Conf. on Blockchain and Cryptocurrency*, pages 1–9, 2020. DOI: 10.1109/ICBC48266.2020.9169454.

[38] Nasir, Qassim, Qasse, Ilham A., Abu Talib, Manar, and Nassif, Ali Bou. Performance analysis of Hyperledger Fabric platforms. *Security and Communication Networks*, pages 1–14, 2018. DOI: 10.1155/2018/3976093.

[39] Neumayr, Bernd, Anderlik, Stefan, and Schrefl, Michael. Towards ontology-based OLAP: Datalog-based reasoning over multidimensional ontologies. In *Int. Conf. on Information and Knowledge Management*, pages 41–48, 2012. DOI: 10.1145/2390045.2390053.

[40] Nguyen, Minh Quang, Loghin, Dumitrel, Tuan, Tien, and Dinh, Anh. Understanding the scalability of Hyperledger Fabric. In *45th International Conference on Very Large Data Bases*, 2019.

[41] Nguyen, Thanh Son Lam, Jourjon, Guillaume, Potop-Butucaru, Maria, and Thai, Kim Loan. Impact of network delays on Hyperledger Fabric. In *INFOCOM 2019 — IEEE Conf. on Computer Communications Workshops 2019*, pages 222–227, 2019. DOI: 10.1109/INFCOMW.2019.8845168.

[42] Niemi, Tapio and Niinimäki, Marko. Ontologies and summarizability in OLAP. In *Proc. of the ACM Symp. on Applied Computing*, pages 1349–1353, 2010. DOI: 10.1145/1774088.1774378.

[43] Okeyo, George, Chen, Liming, Wang, Hui, and Sterritt, Roy. A hybrid ontological and temporal approach for composite activity modelling. In *Proc. of the 11th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications*, pages 1763–1770, 2012. DOI: 10.1109/TrustCom.2012.34.

[44] Parsia, Bijan, Matentzoglu, Nicolas, Gonçalves, Rafael S., Glimm, Birte, and Steigmiller, Andreas. The OWL Reasoner Evaluation (ORE) 2015 competition report. *Journal of Automated Reasoning*, 59(4):455–482, 2017. DOI: 10.1007/s10817-017-9406-8.

[45] Perer, Adam and Shneiderman, Ben. Systematic yet flexible discovery: Guiding domain experts through exploratory data analysis. In *Proc. Int. Conf. on Intelligent User Interfaces*, pages 109–118. ACM Press, 2008. DOI: 10.1145/1378773.1378788.

[46] Pnueli, Amir. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13(1):45–60, 1981. DOI: 10.1016/0304-3975(81)90110-9.

[47] Pongnumkul, Suporn, Siripanpornchana, Chaiyaphum, and Thajchayapong, Suttipong. Performance analysis of private blockchain platforms in varying workloads. In *26th Int. Conf. on Computer Communications and Networks*, pages 1–6, 2017. DOI: 10.1109/ICCCN.2017.8038517.

[48] Prat, Nicolas, Megdiche, Imen, and Akoka, Jacky. Multidimensional models meet the semantic web: Defining and reasoning on OWL-DL ontologies for OLAP. In *Int. Conf. on Information and Knowledge Management, Proc.*, pages 17–24, 2012. DOI: 10.1145/2390045.2390049.

[49] Riboni, Daniele and Bettini, Claudio. OWL 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing*, 7(3):379–395, 2011. DOI: 10.1016/j.pmcj.2011.02.001.

[50] Shalaby, Salma, Abdellatif, Alaa Awad, Al-Ali, Abdulla, Mohamed, Amr, Erbad, Aiman, and Guizani, Mohsen. Performance evaluation of Hyperledger Fabric. In *IEEE Int. Conf. on Informatics, IoT, and Enabling Technologies*, pages 608–613, 2020. DOI: 10.1109/ICIoT48696.2020.9089614.

[51] Sharma, Ankur, Schuhknecht, Felix Martin, Agrawal, Divya, and Dittrich, Jens. How to databasify a blockchain: the case of Hyperledger Fabric. *arxiv.org*, 2018. DOI: 10.48550/arXiv.1810.13177, Preprint.

[52] Streit, Marc, Schulz, Hans-Jörg, Lex, Alexander, Schmalstieg, Dieter, and Schumann, Heidrun. Model-driven design for the visual analysis of heterogeneous data. *IEEE Trans. on Visualization and Computer Graphics*, 18(6):998–1010, 2012. DOI: 10.1109/TVCG.2011.108.

[53] Sukhwani, Harish, Martínez, José M., Chang, Xiaolin, Trivedi, Kishor S., and Rindos, Andy. Performance modeling of PBFT consensus process for permissioned blockchain network (Hyperledger Fabric). In *IEEE Symp. on Reliable Distributed Systems*, pages 253–255, 2017. DOI: 10.1109/SRDS.2017.36.

[54] Sukhwani, Harish, Wang, Nan, Trivedi, Kishor S., and Rindos, Andy. Performance modeling of Hyperledger Fabric (permissioned blockchain network). In *17th IEEE Int. Symp. on Network Computing and Applications*, 2018. DOI: 10.1109/NCA.2018.8548070.

[55] Thakkar, Parth, Nathan, Senthil, and Viswanathan, Balaji. Performance benchmarking and optimizing Hyperledger Fabric blockchain platform. In *26th IEEE Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 264–276, 2018. DOI: 10.1109/MASCOTS.2018.00034.

[56] van der Aalst, Wil et al. Process mining manifesto. In *Lecture Notes in Business Information Processing*, volume 99, pages 169–194, 2012. DOI: 10.1007/978-3-642-28108-2_19.

[57] Wang, Canhui and Chu, Xiaowen. Performance characterization and bottleneck analysis of Hyperledger Fabric. In *Int. Conf. on Distributed Computing Systems*, pages 1281–1286, 2020. DOI: 10.1109/ICDCS47774.2020.00165.

[58] Wang, Shuo. Performance evaluation of Hyperledger Fabric with malicious behavior. In *Lecture Notes in Computer Science*, volume 11521, pages 211–219, 2019. DOI: 10.1007/978-3-030-23404-1_15.

[59] Wirth, Rüdiger. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, volume 1, pages 29–39, 2000. http://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf.

[60] Wun, Alex, Petrovi, Milenko, and Jacobsen, Hans Arno. A system for semantic data fusion in sensor networks. In *Proc. of the 2007 Inaugural Int. Conf. on Distributed Event-Based Systems*, volume 233, pages 75–79, 2007. DOI: 10.1145/1266894.1266907.

[61] Xiao, Ling, Gerth, John, and Hanrahan, Pat. Enhancing visual analysis of network traffic using a knowledge representation. In *IEEE Symp. on Visual Analytics Science and Technology, VAST 2006*, pages 107–114, 2006. DOI: 10.1109/VAST.2006.261436.

[62] Xu, Xiaoqiong, Sun, Gang, Luo, Long, Cao, Huilong, Yu, Hongfang, and V.Vasilakos, Athanasios. Latency performance modeling and analysis for Hyperledger Fabric blockchain network. *Information Processing & Management*, 58(1):102436, 2021. DOI: 10.1016/j.ipm.2020.102436.

[63] Yang, Di, Rundensteiner, Elke A., and Ward, Matthew O. Analysis guided visual exploration of multivariate data. In *VAST IEEE Symp. on Visual Analytics Science and Technology 2007*, pages 83–90, 2007. DOI: 10.1109/VAST.2007.4389000.

[64] Yuan, Pu, Zheng, Kan, Xiong, Xiong, Zhang, Kuan, and Lei, Lei. Performance modeling and analysis of a Hyperledger-based system using GSPN. *Computer Communications*, 153:117–124, 2020. DOI: 10.1016/j.comcom.2020.01.073.

Received 19th December 2021

Acta Cybernetica **26** (2023) 215–241.

Dual Convolutional Neural Network Classifier with Pyramid Attention Network for Image-Based Kinship Verification*

Reza Fuad Rachmadi[ab], I Ketut Eddy Purnama[ac], Supeno Mardi Susiki Nugroho[de], and Yoyon Kusnendar Suprapto[df]

Abstract

A family is the smallest entity that formed the world with specific characteristics. The characteristics of a family are that the member can/may share some similar DNA and leads to similar physical appearances, including similar facial features. This paper proposed a dual convolutional neural network (CNN) with a pyramid attention network for image-based kinship verification problems. The dual CNN classifier is formed by paralleling the FaceNet CNN architecture followed by family-aware features extraction network and three final fully-connected layers. A channel-wise pyramid attention network is added after the last convolutional layers of FaceNet CNN architecture. The family-aware features extraction network is used to learn family-aware features using the SphereFace loss function. The final features used to classify the kin/non-kin pair are joint aggregation features between the pyramid attention features and family-aware features. At the end of the fully connected layer, a softmax loss layer is attached to learn kinship verification via binary classification problems. To analyze the performance of our proposed classifier, we performed experiments heavily on the Family in The Wild (FIW) kinship verification dataset. The FIW kinship verification dataset is the largest dataset for kinship verification currently available. Experiments of the FIW dataset show that our proposed classifier can achieve the highest average accuracy of 68.05% on a single classifier scenario and 68.73% on an ensemble classifier scenario which is comparable with other state-of-the-art methods.

*This research is partially supported by Indonesia Minister of Education, Culture, and Higher Education Grant (under PDUPT scheme) number 930/PKS/ITS/2019 and 1224/PKS/ITS/2020.

[a]Dept. of Computer Engineering and University Center of Excellence on Artificial Intelligence for Healthcare and Society (UCE AIHeS), Institut Teknologi Sepuluh Nopember, Indonesia

[b]E-mail: fuad@its.ac.id, ORCID: 0000-0001-9101-5598

[c]E-mail: ketut@te.its.ac.id, ORCID: 0000-0002-7438-7880

[d]Dept. of Computer Engineering, Institut Teknologi Sepuluh Nopember, Indonesia

[e]E-mail: mardi@its.ac.id, ORCID: 0000-0001-8109-6136

[f]E-mail: yoyon@te.its.ac.id, ORCID: 0000-0003-3149-5088

DOI: 10.14232/actacyb.296355

Keywords: dual CNN classifier, image-based kinship verification, Deep Metric Learning, Channel-wise Pyramid Attention Network

1 Introduction

Humans are unique species in the universe that discriminate by visual appearances, including human faces, fingerprints, retina patterns, and gait. All of those visual appearances are widely used as biometric authentication features of identity. The human faces are a little bit special due to the visual appearances that can be descent from parents or grandparents and can be used to analyze the kinship relationship among people. In this modern era, the camera sensor is widely used to capture images. Many of those images were uploaded to the internet, including photos with human faces and family photos. In recent years, several kinship relationship datasets were formed by researchers to support the development of image-based kinship relationship problems, including KinFaceW-I [26, 27], KinFaceW-II [26, 27], KFVW (Kinship Face Video in The Wild) [54], Cornell KinFace [12], Tri-Subject Kinship [31] and FIW (Family in The Wild) [37, 50, 35]. The dataset is usually formed by crawling well-known families' photos on the web, including actresses and the royal family with clear kinship relationships between the family members.

One of the technologies that provide an opportunity to develop image-based kinship verification problems is the evolution of deep learning methods widely used after Krizhevsky et al. [17] won the ILSVRC 2012 challenges by using a convolutional neural network classifier. After 2012, deep learning is constantly used for a lot of problems and applications, from computer science to remote sensing applications. There are several deep learning approaches for image-based kinship verification problems, including the one described in [20, 21, 11, 50, 8, 32, 35, 33, 30, 36, 53].

In this paper, we proposed a dual convolutional neural network (CNN) classifier with joint features aggregation and a pyramid attention network for image-based kinship verification problems. Our proposed classifier was formed by paralleling the FaceNet CNN architecture [40] and adding two subnetworks, one for family-aware features extraction and one for kin/non-kin classification. Our contributions can be listed as follows.

- We investigated a dual CNN classifier with joint features aggregation for image-based kinship verification problems. The experiments are heavily performed using the FIW dataset [37, 50, 35], which is considered the largest kinship verification dataset currently available.
- We investigated the combination of our proposed classifier with a channel-wise pyramid attention network. The attention network described by Zhao and Wu [58] is adopted with our proposed classifier. Experiments on the FIW dataset show that adding a channel-wise pyramid attention network can improve the classifier's performance.

- For further analysis, we also investigated our proposed classifier with a subset of the FIW dataset, including RFIW'17 [38] and RFIW'18 [35]. The subset of the FIW dataset is used for the competition, which can be compared side-by-side with other methods in the competition.

The rest of the paper is organized as follows. Section 2 discussed several related works on image-based kinship relationship analysis. Our proposed classifier is described in section 3, follows by results and discussion in section 4. Finally, we conclude the experiments in the last section.

2 Related Work

In recent years, there are several works on image-based kinship relationship analysis, including those described in [8, 49, 33, 39, 56, 19]. Dawson et al. [8] reported a performance comparison FSP (From-Same-Photo) classifier on several kinship verification datasets. The FSP classifier is trained on the same photo dataset instead of kinship verification data. The results show that the performance is very good on some kinship verification datasets (some achieved around 90%). Dawson et al. [8] conclude that some kinship verification datasets are not suitable for model development because a lot of the data is taken from the same photo.

Robinson et al. [39] described the RFIW 2020 challenges results with three tasks: kinship verification, tri-subject verification, and search & retrieval of missing children. To create a baseline performance, Robinson et al. [39] use SphereFace CNN classifier [24] which proved to produce high accuracy on face recognition tasks. The baseline performances of the SphereFace classifier are 64% on kinship verification tasks, 68% on tri-subject kinship verification, and mAP of 0.02 on missing children search & retrieval tasks.

Yu et al. [56] proposed a deep fusion siamese network for kinship verification problems. The deep siamese network is used to extract the features of two faces input. The features are fed into a features fusion network before classifying using fully connected layers with a sigmoid activation function at the network's end. Yu et al. [56] perform experiments using several different features fusion mechanisms and two different loss functions, including BCE (Binary Cross Entropy) loss and focal loss. Experiments on RFIW 2020 dataset show that the proposed classifier achieves an average accuracy of 76% on kinship verification problems and 79% on tri-subject kinship verification problems.

A combination of the Young Generation Model with Sparse Discriminative Metric Loss (SDM-Loss) was proposed by Wang et al. [49] for kinship verification problems, especially for parents-child and grandparents-grandchild kinship. The model is based on StarGAN CNN architecture described by Choi et al. [5] and modified the loss with SDM-Loss. Experiments on 5-folds FIW dataset show that ResNet+SDMLoss with an additional young generation model can achieve an average accuracy of 68.68% with siblings and 69.47% without siblings kinship. The testing is divided into two protocols because the proposed classifier uses a young

generation model that may not work properly when combined with siblings kinship that has lower different ages than parent-child or grandparents-grandchild kinship.

Laiadi et al. [19] use Multilinear Side-Information based Discriminant Analysis integrating Within Class Covariance Normalization (MSIDA+WCCN) to train a model for image-based kinship verification problems. The features used by the model are extracted from the fc6 and fc7 layers of four VGG-based CNN that are trained using the ImageNet dataset. The final decision is decided using a simple cosine similarity score between features extracted from two faces using the MSIDA+WCCN model. The proposed model was tested using the KinFaceW dataset and achieved an average accuracy of 87.65% and 87% on the KinFaceW-I and KinFaceW-II datasets.

3 Proposed Classifier

This section describes our proposed classifier, which consists of two different things, the dual CNN classifier with family-aware features and channel-wise pyramid attention network. Figure 1 shows the diagram of our proposed classifier.

3.1 Dual Convolutional Neural Network

The dual CNN classifier of our proposed classifier is formed by paralleling FaceNet CNN architecture which will process for each face image pair. An additional family-aware features extraction network is attached at the end of the classifier, which is adapted from [33]. We use joint features aggregation between pyramid features and family-aware features to improve the classifier's performance. Those joint features aggregation networks proved can improve the classifier's performance in some tasks, including super-resolution tasks [22] and remote sensing image classification [28]. Unlike the dual CNN classifier used in [33], the backbone of our dual CNN classifier weights is not frozen but updated in the training process with a 0.001 times lower learning rate comparing with a fully connected and pyramid attention network. We use three different loss functions that can be computed as follows.

$$L = L_k + \alpha(L_{f1} + L_{f2}) \tag{1}$$

with L_k is the loss function of kin/non-kin classification loss, L_{f1} and L_{f2} is the loss function for learning family-aware features, and α is the contributing factor to the final loss value. We use $\alpha > 1$ for the training process, which will let the classifier learn the family-aware features strongly.

To learn the family-aware features, we use two different deep metric learning widely used for face recognition tasks, including SphereFace [24] and Center Loss [51]. Deep metric learning can be divided into two categories, euclidean metric-based loss [43, 42, 40, 51, 13] and cosine metric-based loss [25, 47, 24, 48, 9]. The SphereFace is deep metric learning that cosine metric-based loss function, which

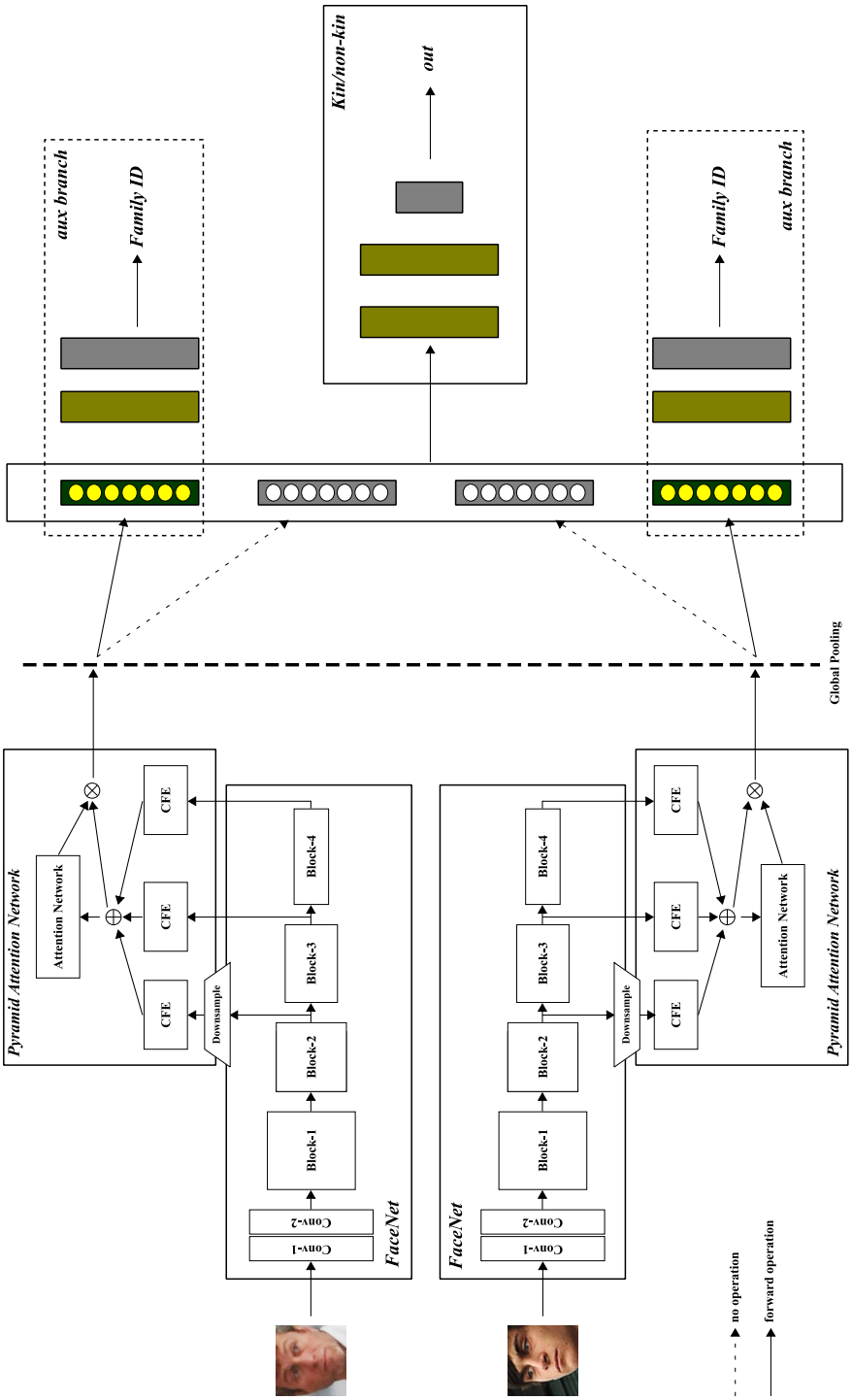


Figure 1: Diagram of our proposed classifier with joint features aggregation and channel-wise pyramid attention network. The face images are taken from the FIW dataset [37, 50, 35].

can be computed as follows.

$$L_a = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{e^{||\mathbf{x}_i||\psi(\theta_{c_i,i})}}{e^{||\mathbf{x}_i||\psi(\theta_{c_i,i})} + f_s(c_i)}\right) \tag{2}$$

$$f_s(c_i) = \sum_{j\neq c_i} e^{||\mathbf{x}_i||\cos(\theta_{c_i,i})} \tag{3}$$

with $\psi(\theta_{c_i,i})$ defined as $\psi(\theta_{c_i,i}) = (-1)^k \cos(m\theta_{c_i,i}) - 2k$, $\theta_{c_i,i} \in \left[\frac{k\phi}{m}, \frac{(k+1)\phi}{m}\right]$, and $k \in [0, m-1]$. We use $m = 4$ to performs the training process as described in the original SphereFace paper [24]. The second deep metric learning used to train our proposed classifier is center loss [51]. The center loss works by minimizing the variation of the intra-class features while trying to separate the features between classes. The loss function for center loss is divided into two functions; the first loss function is used to update the center or centroid of the features, while the second loss function is used to classify the features based on their label. Let $\mathbf{x}_i$ is the extracted features of the last layer of the classifier and $\mathbf{c}_{y_i}$ is the centroid of the features of class y_i-th of the data, the loss function used for updating the center can be computed as follows.

$$L_c = \frac{1}{2}\sum_{i=1}^{N} \left|\left|\mathbf{x}_i - \mathbf{c}_{y_i}\right|\right|_2^2 \tag{4}$$

The efficient way to update the centroid of the features is by analyzing all of the examples and deciding the centroid's shift based on the error produced by the examples. The process is not possible when training the classifier using the mini-batch SGD algorithm. Instead, Wen et al. [51] proposed a joint loss function between softmax and center loss that can be computed as follows.

$$L_f = L_s + \mu L_c \tag{5}$$

$$= -\sum_{i=1}^{m} \log \frac{e^{\mathbf{W}_{\mathbf{y}_i}^T \mathbf{x}_i + b_{\mathbf{y}_i}}}{\sum_j^n e^{\mathbf{W}_j^T \mathbf{x}_i + b_j}} + \frac{\mu}{2}\sum_{i=1}^{m} ||\mathbf{x}_i - \mathbf{c}_i||_2^2 \tag{6}$$

with μ is the contribution of the center loss in the final loss function, and L_s is the softmax loss function. We use $\mu = 0.008$ to performs the training process, which the original authors also recommend.

3.2 Channel-wise Pyramid Attention Network

Attention network is one type of additional network that explores the importance of features on the tasks. The attention network is widely and originally used for natural language processing problems, including that described in [2, 29, 41, 55, 45, 3, 44, 1, 10]. As time goes by, some researchers also tried to implement an attention network for the problem with an image as an input of the classifier,

including that described in [46, 15, 52, 57, 58]. Zhao et al. [58] proposed a pyramid attention network for saliency detection problems. The pyramid attention network consists of two types of attention network, channel-wise attention network and spatial attention network. The channel-wise pyramid attention network computes the importantness of the features per channel, while the spatial attention network computes importantness per feature based on spatial coordinates.

Our proposed classifier adopted the channel-wise pyramid attention network (PAN) described by Zhao et al. [58] and joined the features with family-aware features [33]. Assume that $\mathbf{z} \in \mathbb{R}^{W \times H \times C}$ is the concatenation of multi-level convolutional layer outputs with $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_C]$, C is the total channel number of the features, and $\mathbf{z}_i \in \mathbb{R}^{W \times H}$ is the i-th channel of $\mathbf{z}$, the output of channel-wise attention network can be calculated as follows.

$$A_c(\mathbf{v}, W) = \sigma(f_{c_2}(\delta(f_{c_1}(\mathbf{v}, W_1)), W_2)) \tag{7}$$

with $\mathbf{v} \in \mathbb{R}^C$ is unfold version of $\mathbf{z}$, W is the parameters in the channel-wise attention network, σ is the sigmoid function, δ is the ReLU function, f_{c_1} and f_{c_2} is the fully-connected function. In our implementation, we use the PReLU function [14] instead of the ReLU which used in the original implementation. We change the activation function to match the activation function used in the backbone network (FaceNet architecture). The final features are computed by weighting the features with the output of the channel-wise attention network as follows.

$$\tilde{\mathbf{z}} = \mathbf{z} \cdot A_c(\mathbf{v}, W) \tag{8}$$

The operation is performed channel-wise multiplication using the attention weights.

The features used to calculate the channel-wise attention outputs are extracted using CFE (Context14 aware Features Extraction) module, which is also used in the original pyramid attention network paper [58]. The CFE module consists of four convolutional layers with different kernel size and dilation rates, 1×1 kernel, 3×3 kernel with dilation rates of 3, 3×3 kernel with dilation rates of 5, and 3×3 kernel with dilation rates of 7. The output of the CFE module is the combination of those four convolutional with additional batch normalization and PReLU activation functions. As shown in Figure 1, the output of convolutional blocks 2, 3, and 4 is used to extract pyramid features using the CFE module and combined it to form the final pyramid features. Features extracted from convolutional block two are downsampled to match the resolution of other features.

3.3 Face Segmentation

To ensure that the classifier only learned the appropriate face features, we applied a face parsing/face segmentation of the input faces in the preprocessing step before the training process. We use a face labeling model described in [4], which utilizes the face labeling problem described in [23] and is used to supplying the semantic segmentation for thermal-to-visible image translation using a generative adversarial network. The face parsing model produces eleven labels of face images, including

(a) (b) (c) (d)

Figure 2: The results of the face parsing model that was used in our pre-processing step. (a) face image, (b) the parsing result with background, (c) the parsing result without background, and (d) the final result used in the training process.

background, left eye, right eye, left eyebrow, right eyebrow, nose, upper lips, lower lips, inside the mouth, facial skin, and hair.

In our experiments, we only take pixels that label as non-background (eyes, eyebrows, nose, lips, mouth, facial skin, and hair) for the training process. Figure 2 shows the face parsing process and removing the background labeled pixels before saved the final images for the training process (as showed in Figure 2-(d)). By using the preprocessing face images, our proposed classifier can achieve a good validation accuracy comparing without the face parsing preprocessing process. The face segmentation preprocessing process is only applied in the training process.

4 Results and Discussion

To evaluate our proposed classifier, we performed a detailed analysis using the FIW dataset [37, 50, 35] and Caffe deep learning framework [16]. We also performed experiments using only a family-aware CNN classifier [33] and several ensemble configurations. Figure 3 shows the flow of the kinship verifiation experiments for our proposed classifier.

4.1 FIW Dataset

The FIW dataset [37, 50, 35] is currently the largest kinship verification dataset and proved to be a challenging problem. The FIW dataset consists of 11,932 face images covering around 1,000 families with eleven different kinship relationship types. The eleven kinship relationship can be divided into three categories, same generation kinship (siblings, brother, and sister), first-generation kinship (mother-son, mother-daughter, father-son, and father-daughter), and second-generation kinship (grand mother-grand son, grand mother-grand daughter, grand father-grand son, and grand father-grand daughter). Figure 4 shows several examples of face images for each kinship category on the FIW dataset. As shown in Figure 4, higher generation kinship may reduce the facial features similarity which reasonable due to combination of DNA from grand parent to parent to grand child. There are several different split configurations (training and testing list) of the FIW dataset. This paper uses three different configurations, including the 5-folds configurations

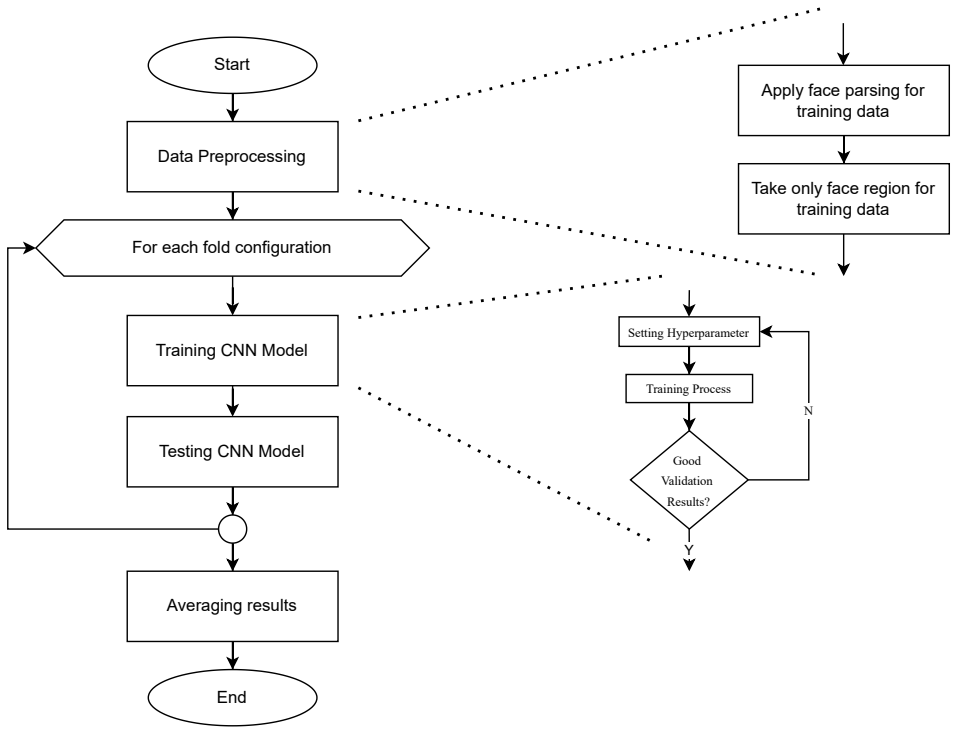


Figure 3: The main flow of our experiments using the FIW dataset.

(no overlapped family between folds), RFIW 2017 challenge, and RFIW 2018 challenge. We heavily perform the experiments using the 5-folds configuration before use RFIW 2017 and RFIW 2018 split configuration.

4.2 Experiments Setup

Implementation Detail. We use four different classifier configurations of our dual CNN classifier with a pyramid attention network. All approaches are based on FaceNet CNN architecture, and final features are constructed by combining family-aware features with pyramid attention features. Each classifier can be described as follows.

- **DFaceNet-FC512-CAtt**. Dual FaceNet classifier combined with 512 family-aware features learned using SphereFace Loss function [24] and channel-wise attention network (CAtt). The total features used for the final fully-connected layers are 896 features with 512 family-aware features and 384 features from the pyramid attention network.
- **DFaceNet-FC1K-CAtt** and **DFaceNet-FC2K-CAtt**. The classifier uses the same configuration as the DFaceNet-FC512-CAtt classifier but with a

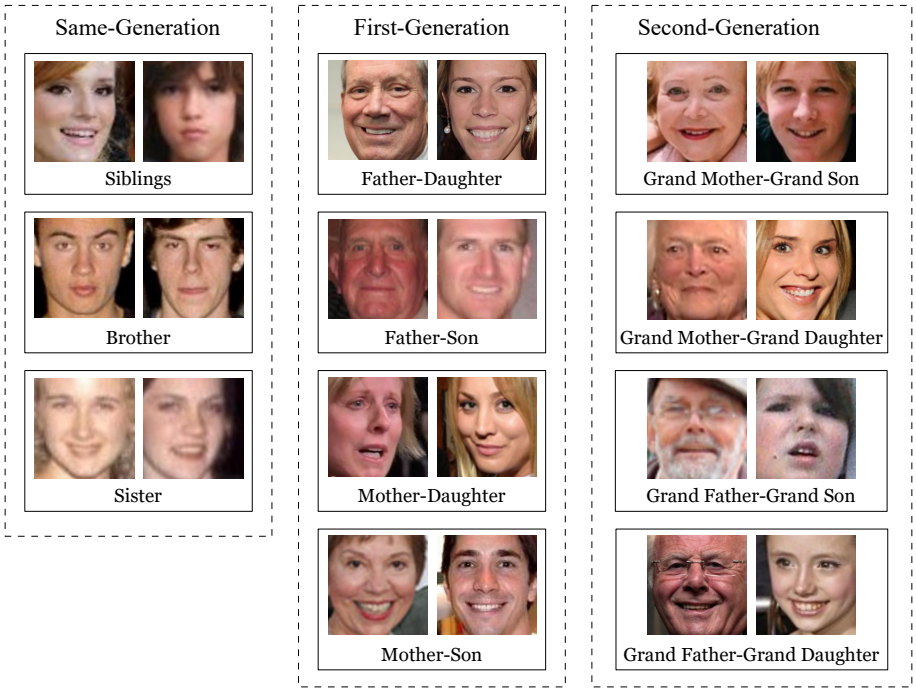


Figure 4: Kinship relationship categories in the FIW dataset and their examples pair of each category [37, 50, 35].

different number of family-aware features, 1,024 and 2,048. The total number of final features for DFaceNet-FC1K-CAtt and DFaceNet-FC2K-CAtt is 1,408 and 2,432 features, respectively.

- **DFaceNet-FC512-ASCL-CAtt**. Similar to DFaceNet-FC512-CAtt but with two different family-aware features, features learned using SphereFace Loss function and features learned using Center Loss function. The final features are 1,024 family-aware features (from two different family-aware branches) and 384 pyramid attention features.

All of the classifiers use a CPFE network with four different atrous convolutions, 1×1 kernel with dilation rate of 1 and 3×3 kernel with dilation rate of 3, 5, and 7. The CPFE network is attached after each output of blocks 2 to 4, and the pyramid features are constructed by combining the output of all CPFE networks.

Training Process. The training process is done for ten epochs using NAG (Nesterov Accelerated Gradient) training algorithm. The learning rate is initialized at 0.01 with a polynomial reducing policy and additional clipping gradient method to reduce the exploding gradient problem, especially in the first couple of epoch. We reduce the learning rate by 0.001 factor for the backbone network to preserve the classifier's ability to extract face features. The input images are resized to 120×120,

followed by random cropping using 112×112 resolution and data normalization before the training process.

Testing Process. In the testing process, we use multi-resolution approaches by classifying the input image using several different input resolutions, including 115×115, 118×118, 122×122, and 128×128. Each resolution is cropped into ten different crops (center, left top, left bottom, right top, right bottom, and their respective mirror version of the crops) with a resolution of 112×112. After the pre-processing process, the testing process performs a classification using 40 crops, and the final classification score is computed by averaging the score of all crops. We performed ensemble testing by using a simple average ensemble mechanism, which proved to improve the classifier's performance by around 1-2%.

4.3 Results and Discussion

The results are divided into four different independent experiments, which are detailed discussed in each sub-section. We added one additional preliminary experiment using the FA-CNN classifier [33], which was used as the basis for our proposed classifier.

4.3.1 Preliminary Results

In the preliminary experiments, we use Dual FaceNet-FA (Family-Aware CNN) [33] with the SphereFace Loss function to learn the family-aware features. We use different training scenarios, which are not time-consuming, as reported in the original paper. The training process is done with the same hyperparameter setting as described in the experiment's setup. Lambda $\lambda = 10$ is used for the SphereFace Loss function, which in the original paper suggested choosing a small lambda value (e.g. 10 or 5) to compensate for the original softmax loss function. Table 1 shows the result of the preliminary experiments using the 5-folds FIW dataset and three different classifier configurations. As shown in Table 1, the average accuracy of the classifier is similar to the one reported in [33], although we use a different training scenario. The second-generation kinship relationship still produces the lowest accuracy due to the limited data available in the dataset.

4.3.2 5-Folds Configuration

After preliminary experiments, we conducted the experiments using a Dual FaceNet classifier with family-aware features and channel-based pyramid attention network features. Four different classifiers along with ensemble configuration were used to perform the experiments. Table 2 shows the results of the Dual FaceNet classifier with family-aware features and channel-wise pyramid attention network features with average accuracy ranged from 67.80% to 68.05%. As shown in Table 2, the best performance of the single classifier is achieved using the DFaceNet-FC1K-CAtt classifier with an average accuracy of 68.05%. The second-generation kinship verification seems still the hardest case for the classifier with average accuracy

Table 1: Verification results (%) on FIW dataset for Dual FaceNet classifier using 5-fold configuration (no family overlapped between folds).

#	Method	siblings			parent-child				grandparent-grandchild				Avg
		ss	bb	sibs	fd	fs	md	ms	gfgd	gfgs	gmgd	gmgs	
1.	DFaceNet-FC512-λ10	74.8	69.0	70.3	68.8	68.1	71.8	70.2	**62.0**	**62.9**	**62.8**	64.3	**67.78**
2.	DFaceNet-FC1K-λ10	74.8	68.7	70.4	69.0	**71.9**	70.3	68.0	**62.0**	62.7	61.3	**64.5**	67.65
3.	DFaceNet-FC2K-λ10	**75.3**	**69.5**	**70.4**	**69.1**	68.4	**72.4**	**70.6**	61.9	61.8	61.2	64.1	67.75

Table 2: Verification results (%) on FIW dataset for Dual FaceNet with family-aware features and channel-based attention network using 5-fold configuration (no family overlapped between folds).

#	Method	siblings			parent-child				grandparent-grandchild				Avg
		ss	bb	sibs	fd	fs	md	ms	gfgd	gfgs	gmgd	gmgs	
1.	DFaceNet-FC512-CAtt	75.0	69.4	70.4	68.9	68.0	71.6	70.4	63.3	61.5	62.9	63.8	67.80
2.	DFaceNet-FC1K-CAtt	75.5	69.8	70.5	69.4	68.1	72.0	70.7	62.6	62.1	63.1	64.2	68.05
3.	DFaceNet-FC2K-CAtt	75.7	69.6	70.6	69.2	68.2	72.1	70.4	62.6	61.5	62.1	63.7	67.85
4.	DFaceNet-FC512-CL-CAtt	75.7	69.9	71.2	69.0	68.4	71.9	70.3	62.2	62.9	62.1	63.9	67.98
5.	Ensemble 1 + 4	75.9	70.0	71.2	69.5	68.7	72.4	71.0	**63.4**	**63.0**	62.9	63.7	68.38
6.	Ensemble 2 + 4	76.1	70.3	71.3	69.8	68.8	72.7	71.3	63.0	62.6	**63.5**	64.0	68.55
7.	Ensemble 1 + 2 + 4	76.0	**70.3**	71.4	69.9	69.0	72.8	71.4	63.2	62.5	63.4	64.0	68.59
8.	Ensemble All	**76.2**	**70.3**	**71.5**	**70.1**	**69.0**	**73.0**	**71.5**	63.4	62.7	63.5	**64.4**	**68.73**

(a) makeup

(b) large age difference

(c) color

Figure 5: Examples of the correct classification (left) and incorrect classification (right) on sister kinship relationship using same person pair.

ranged from 62% to 64%. The highest accuracy appears on sister and mother-daughter kinship type, which may be supported by the fact that sister and mother-daughter may like the same favourite makeup style and may contribute to the training process.

We take a quick analysis for the sister kinship relationship category, and Figure 5 shows the pair that correctly classify (left side) and incorrectly classify (right side). Three different factors affected the classification process; the first one is the makeup used in the photo (Figure 5-(a)), large-age difference (Figure 5-(b)), and color (Figure 5-(c)). We believe that those three factors also affected other kinship relationship categories. That is why the sister kinship relationship type achieved the highest average accuracy compared with other types of kinship.

To further improve the classifier's performance, we also conducted the testing process using four ensemble configurations, as shown in Table 2. As shown in Table 2, the ensemble configuration can improve the classifier's performance by around 0.5-0.8% compared with the single classifier configuration. As predicted, the second generation kinship relationship categories still produce the lowest average accuracy. Still, it is relatively higher compared with the single classifier results except for the grand mother-grand son kinship category. The best average accuracy of the ensemble classifier is 68.73% using Ensemble All (four of the DFaceNet classifiers).

Figure 6 shows the ROC curve plot of eleven kinship relationship categories from three different classifiers, the DFaceNet-FC512 classifier, DFaceNet-FC1K-CAtt classifier, and Ensemble All configuration. The AUC score is also included in the graph to provided insight information regarding the classifier. As shown in Figure 6, the ensemble configuration provides around 0.01 increase on the AUC score. The ROC curve of second-generation kinship relationship categories is not smooth with a lot of jigsaw-like lines, especially on grand father-grand daughter kinship.

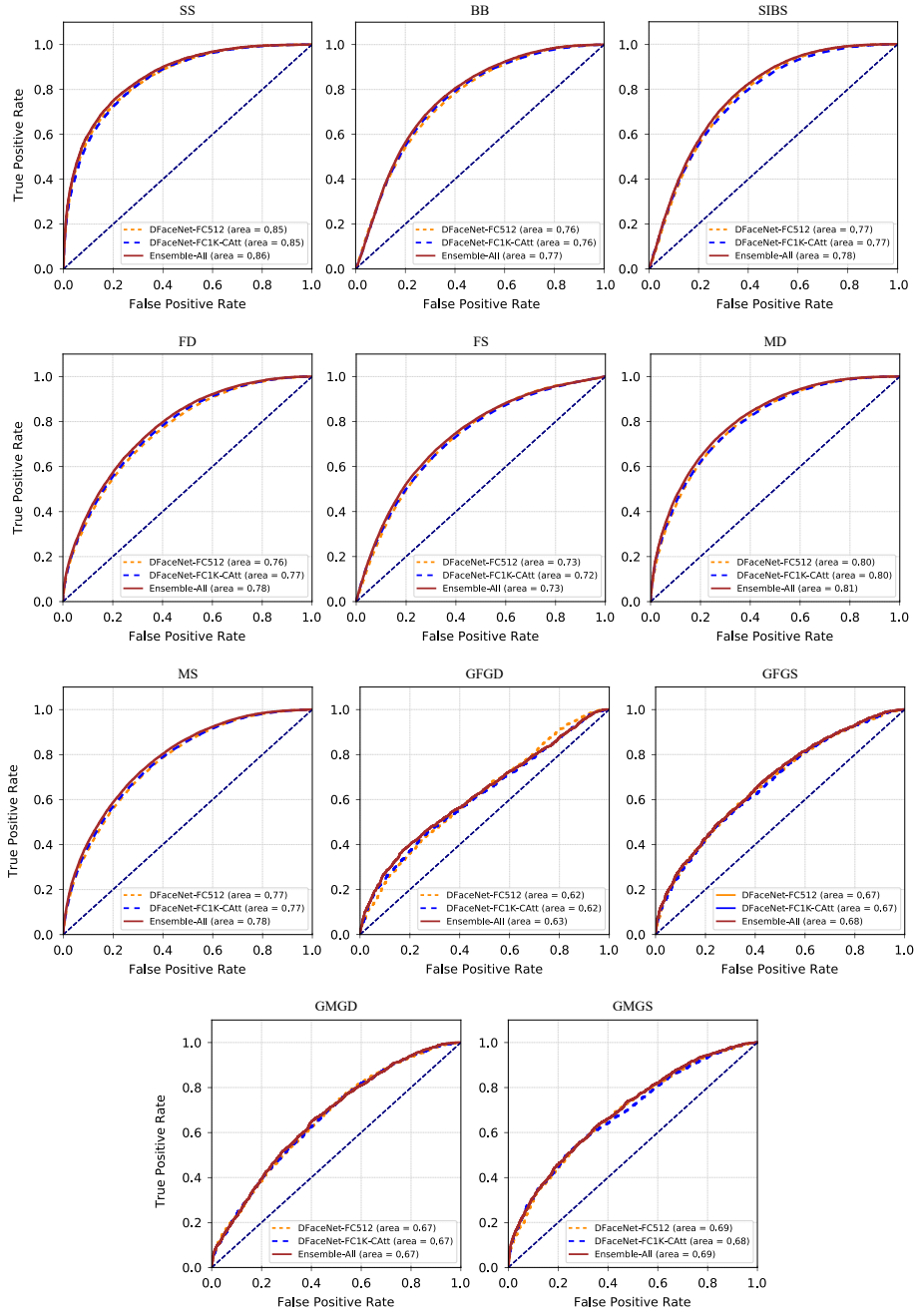


Figure 6: ROC curve of three different classifiers, the DFaceNet-FC512, DFaceNet-FC1K-CAtt, and Ensemble All, for 5-folds split configuration on FIW dataset.

Table 3: Comparison of our proposed classifier with several other methods on 5-folds FIW dataset.

No.	Method	Siblings	Parent Child	Grand Grand	Avg. All
1.	SphereFace [35]	73.15	69.76	65.60	69.18
2.	SDMLoss [49]	74.11	69.08	64.22	68.68
3.	DML[1] [50]	75.27	70.05	65.89	68.79
4.	Dual VGG-Face [34]	69.43	66.65	61.37	65.49
5.	FA-CNN [33]	73.64	71.12	62.93	68.84
6.	**Our method**	**72.68**	**70.94**	**63.53**	**68.73**

The AUC (area under the curve) on the ROC curve shows that the worst performance occurs in the grand father-grand daughter class and the best performance is occurs in the sister class.

Comparison with state-of-the-art (SOTA). We listed several different other methods that use 5-folds FIW dataset. In the early FIW dataset, the 5-folds configuration consists of only nine instead of eleven kinship categories [50]. Although Wang et al. [50] use different 5-folds configurations, we still included the results for information preservation because we cannot recreate the experiments due to no available information regarding the split configuration. Table 3 shows the comparison between our proposed classifier with several other methods on the 5-folds FIW dataset. We also include the average accuracy of each generation (siblings, parent-child, and grand parent-grand child) to provide more information regarding the classifier's performance on different generations.

4.3.3 RFIW'17

We use the RFIW2017 challenge split configurations to perform similar experiments as in the 5-folds experiments to make more comparisons. We use the same hyperparameter and epoch to perform the training process and tested using the validation dataset only because submission to the challenge website is already closed by the organizer. Table 4 shows the results of our proposed classifier on the RFIW'17

[1]The 5-folds dataset is different with nine kinship relationship categories instead of eleven.

Table 4: Verification results (%) on RFIW'17 validation dataset for Dual FaceNet with family-aware features and channel-based attention network.

#	Method	siblings			parent-child				Avg
		ss	bb	sibs	fd	fs	md	ms	
1.	DFaceNet-FC512-CAtt	75.5	73.4	72.5	67.5	67.7	70.8	70.3	71.16
2.	DFaceNet-FC1K-CAtt	77.9	71.6	71.9	67.6	67.9	70.6	70.6	71.20
3.	DFaceNet-FC2K-CAtt	76.0	72.1	71.6	67.6	68.0	70.6	69.8	70.86
4.	DFaceNet-FC512-CL-CAtt	76.4	70.9	70.9	66.7	67.3	69.9	68.6	70.15
5.	Ensemble 1 + 4	76.4	73.5	73.0	68.2	68.3	71.2	70.8	71.68
6.	Ensemble 2 + 4	**78.3**	72.6	72.2	68.3	68.5	70.8	70.7	71.66
7.	Ensemble 1 + 2 + 4	77.6	73.7	73.3	68.5	68.7	71.5	71.5	72.17
8.	Ensemble All	77.5	**73.9**	**73.5**	**69.1**	**69.2**	**71.8**	**71.7**	**72.44**

Table 5: Verification results (%) on RFIW'18 validation dataset for Dual FaceNet with family-aware features and channel-based attention network.

#	Method	siblings			parent-child				grandparent-grandchild				Avg
		ss	bb	sibs	fd	fs	md	ms	gfgd	gfgs	gmgd	gmgs	
1.	DFaceNet-FC512-CAtt	72.3	75.7	76.3	68.8	67.9	70.4	71.1	55.2	62.9	59.0	59.3	67.22
2.	DFaceNet-FC1K-CAtt	73.2	76.2	76.6	68.8	68.5	70.4	71.1	53.4	64.4	58.7	59.8	67,42
3.	DFaceNet-FC2K-CAtt	72.8	**76.5**	77.4	69.3	68.8	70.7	71.6	54.9	**65.5**	57.6	59.2	67.69
4.	DFaceNet-FC512-CL-CAtt	72.9	76.0	76.9	68.6	68.2	70.2	71.0	**56.0**	63.5	58.5	59.2	67.41
5.	Ensemble 1 + 4	73.2	76.2	77.5	69.4	68.5	71.1	71.8	55.6	63.7	57.9	59.8	67.73
6.	Ensemble 2 + 4	**73.6**	76.3	77.4	69.3	68.9	71.0	71.7	54.9	64.4	**59.1**	59.4	67.84
7.	Ensemble 1 + 2 + 4	73.6	76.3	77.4	69.7	68.9	71.3	72.0	54.8	64.7	58.6	**59.8**	67.96
8.	Ensemble All	73.5	76.4	**77.8**	**69.8**	**69.1**	**71.4**	**72.2**	55.3	64.4	58.3	**59.8**	**68.05**

dataset. As shown in Table 4, the best performance of single classifier configuration is achieved using the DFaceNet-FC1K-CAtt classifier with an average accuracy of 71.20%. The ensemble configuration is improved by around 0.5-1.0%, and the best performance is achieved using Ensemble All classifier with an average accuracy of 72.44%.

Same with previous experiments, we also plot the ROC curve of each kinship relationship category. Figure 7 shows the ROC curve of three different classifier configurations, including DFaceNet-FC512-CAtt, DFaceNet-FC1K-CAtt, Ensemble-1-2-4, and Ensemble All. As shown in Figure 7, the ROC analysis shows that our proposed classifier performs well with an AUC score of more than 80% except for the father-son and father-daughter kinship relationship. The same AUC score improvement of 0.01 as in the 5-folds experiments also occurs in the RFIW'17 experiments.

Table 6 shows the comparison of our proposed classifier with other methods on the RFIW'17 dataset. Unfortunately, we can also provide the accuracy on the validation set instead of the testing set because the organizer already close the submission server, and we don't have any annotation on the testing set. As shown in Table 6, our proposed classifier is comparable with other methods. We are aware that our proposed classifier does not produce the highest accuracy. Still, in

Table 6: Comparison of our proposed classifier with several other methods on RFIW'17 dataset (average accuracy of each category.

No.	Method	Siblings	Parent Child	Avg. All
1.	KinNet [21]	75.07	74.68	74.85
2.	AdvNet [11]	73.00	68.46	70.41
3.	LPQ-SIEDA [18]	54.53	55.01	54.81
4.	Multi-Set Learning [7]	63.68	62.66	63.10
5.	Parallel SPCNN [32]	62.01	60.81	61.33
6.	FA-CNN [33]	74.52	70.79	72.39
7.	**Our method**[2]	75.02	70.50	72.44

[2]The average accuracy is based on validation set instead of testing set

our understanding, the KinNet approaches [21] use a deeper and bigger classifier, which is natural will have more accuracy than our approaches.

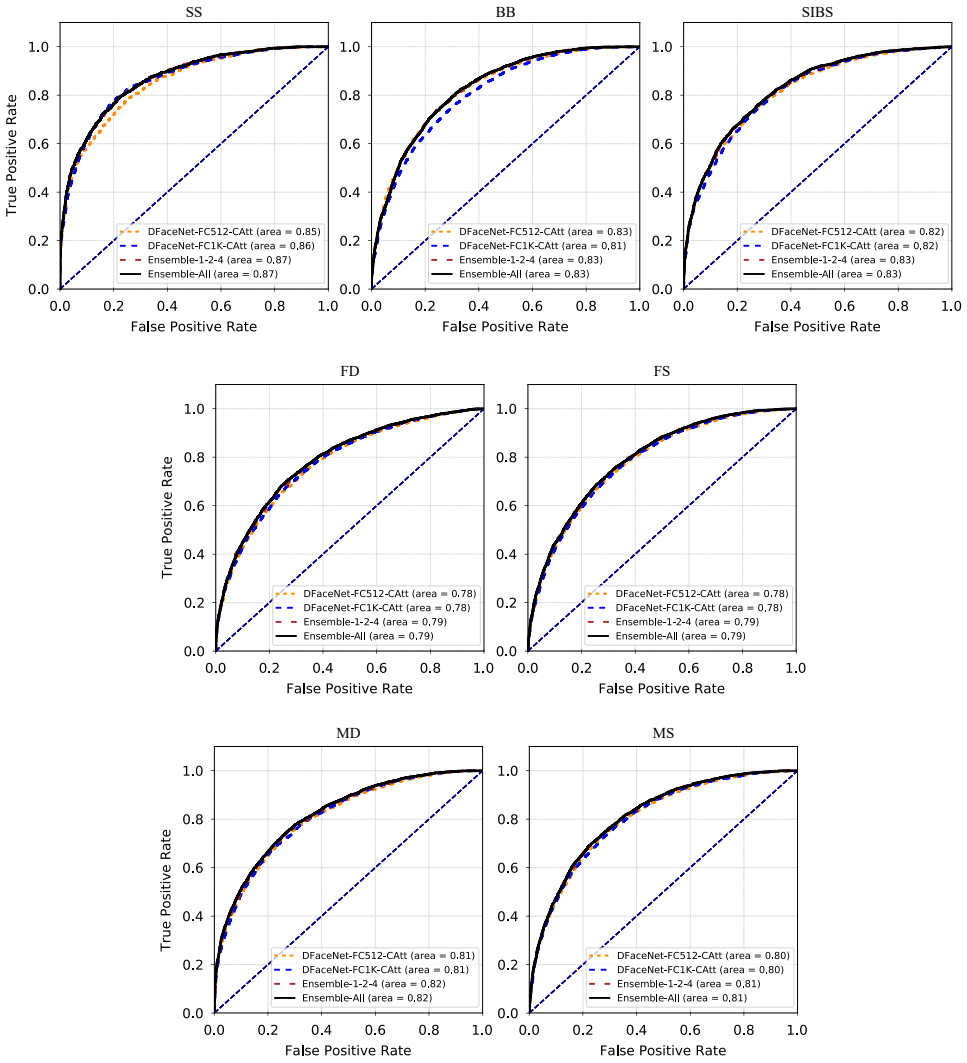


Figure 7: ROC curve of three different classifiers, the DFaceNet-FC512-CAtt, DFaceNet-FC1K-CAtt, Ensemble-1-2-4, and Ensemble All, for the RFIW'17 dataset.

4.3.4 RFIW'18

The last experiment is conducted using the RFIW'18 dataset. The RFIW'18 dataset is a subset of the FIW dataset used for the RFIW challenge 2018 and consists of the same number of kinship relationships as the 5-folds configuration. Same as the previous experiments, the same hyperparameter values were used to perform the experiments. Table 5 shows the results of the experiments using four different single classifiers along with four ensemble configurations. As shown in Table 5, the best single classifier performance is achieved using DFaceNet-FC2K-CAtt with an average accuracy of 67.69%. By using ensemble configuration, the classifier's performance is slightly improved by around 0.5%, with the best average accuracy of 68.05%. As we expected, the worst performance of the proposed classifier is on second generation relationship categories which also occurs in the previous experiments. The difference between RFIW'18 with two previous experiments is that the best performance is not occurring in the sister kinship category but in siblings kinship. We believe that those phenomena occur because the dataset's face images composition may consist of more face pairs with large-gap age.

Figure 8 shows the ROC curve of three different proposed classifiers, including DFaceNet-FC512-CL-CAtt, DFaceNet-FC2K-CAtt, and Ensemble-All. As shown in Figure 8, all classifier configurations do not perform well on the grand father-grand daughter and grand mother-grand daughter category. Same as in the 5-folds experiments, the best performance occurs in the same generation kinship relationships. According to Figure 8, the ensemble configuration can improve the AUC score by around 0.01 on all kinship relationship categories.

Table 7 shows the comparison of our proposed classifier with other methods

Table 7: Comparison of our proposed classifier with several other methods on RFIW'18 dataset (average accuracy of each category.

No.	Method	Siblings	Parent Child	Grand Grand	Avg. All
1.	Group #1 [6]	71.67	70.61	63.17	68.20
2.	Group #2	67.53	62.82	58.38	62.44
3.	Group #3	66.75	62.65	58.87	62.40
5.	FA-CNN [33]	70.34	68.54	62.83	66.96
6.	**Our method**	**70.71**	**69.51**	**61.62**	**66.97**

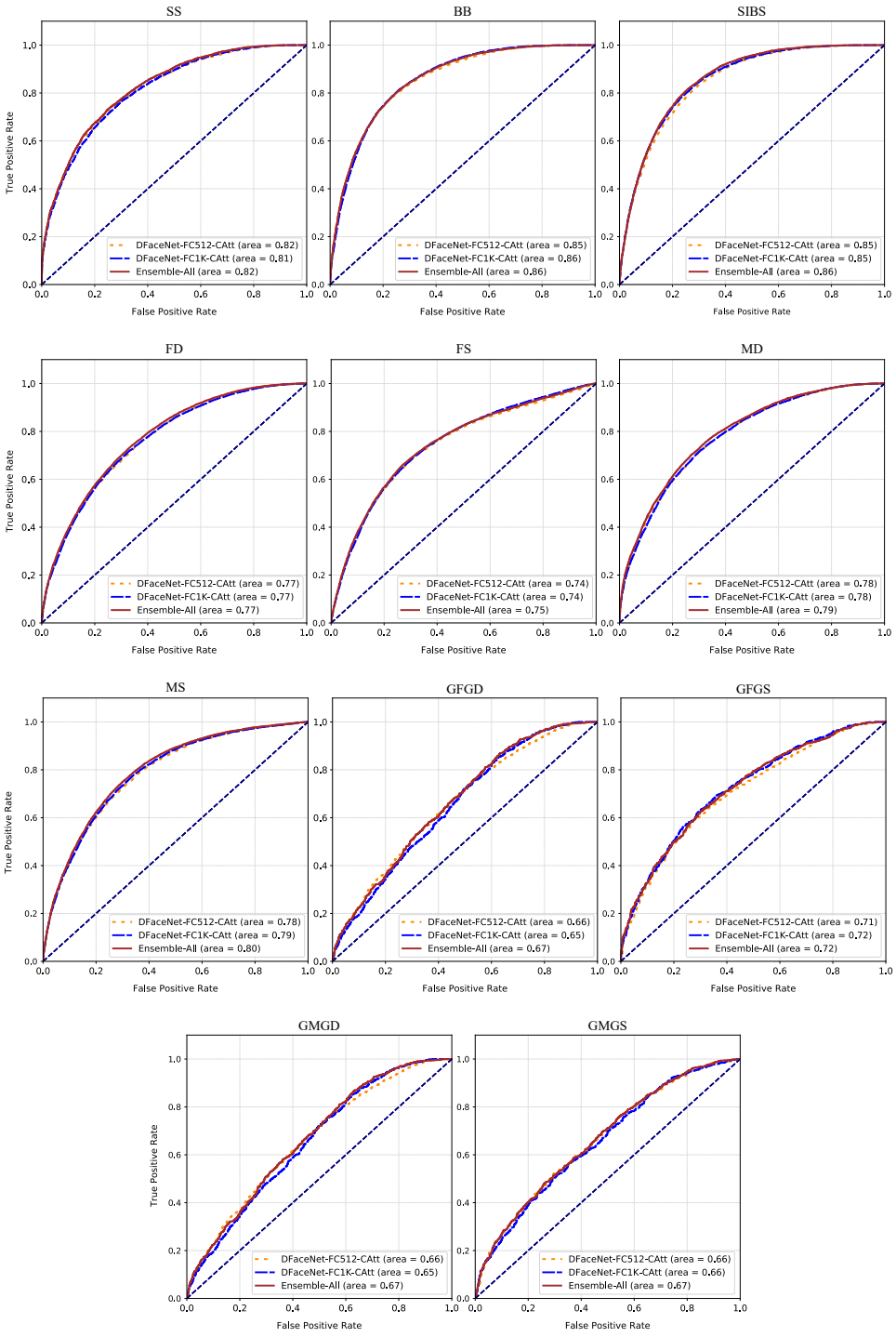


Figure 8: ROC curve of three different classifiers, the DFaceNet-FC512-CL-CAtt, DFaceNet-FC2K-CAtt, and Ensemble All, for RFIW'18 dataset.

on the RFIW'18 dataset. We took the three top participants with the highest performance on the RFIW'18 competition. Unfortunately, the method used by Group #2 and #3 is not published yet. As shown in Table 7, our proposed classifier can achieve an average accuracy of 66.97% and ranked the second-highest performance on the RFIW'18 dataset. Compared with the FA-CNN classifier, the proposed classifier produces a similar performance. Still, the performance per generation shows that the pyramid attention network can improve the performance on same and first-generation kinship relationships while decreasing the performance on second-generation kinship relationships.

5 Conclusion

We present our proposed classifier that combined FaceNet CNN architecture originally used for face recognition with pyramid attention network to solve the kinship verification problem. Our proposed classifier was formed by parallelling the FaceNet CNN architecture and adding family-aware features and a pyramid attention network. The final features were constructed by combining pyramid attention features and family-aware features and fed the features into three fully connected layers to perform the verification tasks. Experiments on three different subsets of the FIW dataset show that the proposed classifier can achieve good accuracy and is comparable with the state-of-the-art classifier on the FIW dataset. The proposed classifier achieves an average accuracy of 69.73% on the 5-folds RFIW dataset, 72.44% on the RFIW'17 dataset, and 66.97% on the RFIW'18 dataset.

For further study, experiments using several different CNN architectures (including non-face recognition architecture) with pyramid attention networks are demanding to show which CNN architectures perform best for image-based kinship verification. The second-generation kinship type may need to be solitary experimented due to lower facial features matched between the pair. Other concerns worth analyzing are the relation between each region of the face for kinship verification problems (e.g., eyes, lips, nose, etc.).

References

[1] Ambartsoumian, Artaches and Popowich, Fred. Self-Attention: A better building block for sentiment analysis neural network classifiers. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 130–139, 2018. DOI: 10.48550/arXiv.1812.07860.

[2] Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015. DOI: 10.48550/arXiv.1409.0473.

[3] Britz, Denny, Goldie, Anna, Luong, Minh-Thang, and Le, Quoc. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451, 2017. DOI: 10.48550/arXiv.1703.03906.

[4] Chen, Cunjian and Ross, Arun. Matching thermal to visible face images using a semantic-guided generative adversarial network. In *IEEE International Conference on Automatic Face & Gesture Recognition*, 2019. DOI: 10.1109/FG.2019.8756527.

[5] Choi, Yunjey, Choi, Minje, Kim, Munyoung, Ha, Jung-Woo, Kim, Sunghun, and Choo, Jaegul. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. DOI: 10.1109/CVPR.2018.00916.

[6] Dahan, Eran and Keller, Yosi. SelfKin: Self adjusted deep model for kinship verification. arXiv Preprint, 2018. DOI: 10.48550/arXiv.1809.08493.

[7] Dahan, Eran, Keller, Yosi, and Mahpod, Shahar. Kin-verification model on FIW dataset using multi-set learning and local features. In *Proceedings of the 2017 Workshop on Recognizing Families In the Wild*, RFIW '17, pages 31–35, New York, NY, USA, 2017. ACM. DOI: 10.1145/3134421.3134423.

[8] Dawson, Mitchell, Zisserman, Andrew, and Nellåker, Christoffer. From same photo: Cheating on visual kinship challenges. In *Asian Conference on Computer Vision*, pages 654–668. Springer, 2018. DOI: 10.1007/978-3-030-20893-6_41.

[9] Deng, Jiankang, Guo, Jia, Xue, Niannan, and Zafeiriou, Stefanos. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. DOI: 10.1109/CVPR.2019.00482.

[10] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. DOI: 10.48550/arXiv.1810.04805.

[11] Duan, Qingyan and Zhang, Lei. AdvNet: Adversarial contrastive residual net for 1 million kinship recognition. In *Proceedings of the 2017 Workshop on Recognizing Families In the Wild*, RFIW '17, pages 21–29, New York, NY, USA, 2017. ACM. DOI: 10.1145/3134421.3134422.

[12] Fang, R., Tang, K. D., Snavely, N., and Chen, T. Towards computational models of kinship verification. In *IEEE International Conference on Image Processing*, pages 1577–1580, 2010. DOI: 10.1109/ICIP.2010.5652590.

[13] Ge, Weifeng, Huang, Weilin, Dong, Dengke, and Scott, Matthew R. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285, 2018. DOI: 10.1007/978-3-030-01231-1_17.

[14] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. DOI: 10.1109/ICCV.2015.123.

[15] Jetley, Saumya, Lord, Nicholas A., Lee, Namhoon, and Torr, Philip H. S. Learn to pay attention. arXiv Preprint, 2018. DOI: 10.48550/ARXIV.1804.02391.

[16] Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, Jonathan, Girshick, Ross, Guadarrama, Sergio, and Darrell, Trevor. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, 2014. DOI: 10.1145/2647868.2654889.

[17] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[18] Laiadi, Oualid, Ouamane, Abdelmalik, Benakcha, Abdelhamid, and Taleb-Ahmed, Abdelmalik. RFIW 2017: LPQ-SIEDA for large scale kinship verification. In *Proceedings of the 2017 Workshop on Recognizing Families In the Wild*, RFIW '17, pages 37–39, New York, NY, USA, 2017. ACM. DOI: 10.1145/3134421.3134426.

[19] Laiadi, Oualid, Ouamane, Abdelmalik, Benakcha, Abdelhamid, Taleb-Ahmed, Abdelmalik, and Hadid, Abdenour. Multi-view deep features for robust facial kinship verification. In *15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 877–881. IEEE, 2020. DOI: 10.1109/FG47880.2020.00118.

[20] Li, Lei, Feng, Xiaoyi, Wu, Xiaoting, Xia, Zhaoqiang, and Hadid, Abdenour. Kinship verification from faces via similarity metric based convolutional neural network. In *International Conference on Image Analysis and Recognition*, pages 539–548. Springer, 2016. DOI: 10.1007/978-3-319-41501-7_60.

[21] Li, Yong, Zeng, Jiabei, Zhang, Jie, Dai, Anbo, Kan, Meina, Shan, Shiguang, and Chen, Xilin. KinNet: Fine-to-coarse deep metric learning for kinship verification. In *Proceedings of the 2017 Workshop on Recognizing Families In the Wild*, RFIW '17, pages 13–20, New York, NY, USA, 2017. ACM. DOI: 10.1145/3134421.3134425.

[22] Liu, Jie, Zhang, Wenjie, Tang, Yuting, Tang, Jie, and Wu, Gangshan. Residual feature aggregation network for image super-resolution. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2359–2368, 2020. DOI: 10.1109/CVPR42600.2020.00243.

[23] Liu, Sifei, Yang, Jimei, Huang, Chang, and Yang, Ming-Hsuan. Multi-objective convolutional learning for face labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3451–3459, 2015. DOI: 10.1109/CVPR.2015.7298967.

[24] Liu, Weiyang, Wen, Yandong, Yu, Zhiding, Li, Ming, Raj, Bhiksha, and Song, Le. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017. DOI: 10.1109/CVPR.2017.713.

[25] Liu, Weiyang, Wen, Yandong, Yu, Zhiding, and Yang, Meng. Large-margin softmax loss for convolutional neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 507–516, 2016.

[26] Lu, Jiwen, Hu, Junlin, Zhou, Xiuzhuang, Shang, Yuanyuan, Tan, Yap-Peng, and Wang, Gang. Neighborhood repulsed metric learning for kinship verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2594–2601. IEEE, 2012. DOI: 10.1109/CVPR.2012.6247978.

[27] Lu, Jiwen, Zhou, Xiuzhuang, Tan, Yap-Pen, Shang, Yuanyuan, and Zhou, Jie. Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):331–345, 2013. DOI: 10.1109/TPAMI.2013.134.

[28] Lu, Xiaoqiang, Sun, Hao, and Zheng, Xiangtao. A feature aggregation convolutional neural network for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(10):7894–7906, 2019. DOI: 10.1109/TGRS.2019.2917161.

[29] Luong, Minh-Thang, Pham, Hieu, and Manning, Christopher D. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015. DOI: 10.48550/arXiv.1508.04025.

[30] Ma, Jiayi, Jiang, Xingyu, Fan, Aoxiang, Jiang, Junjun, and Yan, Junchi. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1):23–79, 2021. DOI: 10.1007/s11263-020-01359-2.

[31] Qin, X., Tan, X., and Chen, S. Tri-subject kinship verification: Understanding the core of a family. *IEEE Transactions on Multimedia*, 17(10):1855–1867, 2015. DOI: 10.1109/TMM.2015.2461462.

[32] Rachmadi, Reza Fuad and Purnama, I Ketut Eddy. Paralel spatial pyramid convolutional neural network untuk verifikasi kekerabatan berbasis citra wajah.

Jurnal Teknologi dan Sistem Komputer, 6(4):152–157, 2018. DOI: 10.14710/jtsiskom.6.4.2018.152-157.

[33] Rachmadi, Reza Fuad, Purnama, I Ketut Eddy, Nugroho, Supeno Mardi Susiki, and Suprapto, Yoyon Kusnendar. Family-aware convolutional neural network for image-based kinship verification. *International Journal of Intelligent Engineering and Systems*, 13(6):20–30, 2020. DOI: 10.22266/ijies2020.1231.03.

[34] Rachmadi, Reza Fuad, Purnama, I Ketut Eddy, Nugroho, Supeno Mardi Susiki, and Suprapto, Yoyon Kusnendar. Image-based kinship verification using dual VGG-Face classifier. In *IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pages 123–128. IEEE, 2021. DOI: 10.1109/IoTaIS50849.2021.9359720.

[35] Robinson, J. P., Shao, M., Wu, Y., Liu, H., Gillis, T., and Fu, Y. Visual kinship recognition of families in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2624–2637, 2018. DOI: 10.1109/TPAMI.2018.2826549.

[36] Robinson, Joseph P, Shao, Ming, and Fu, Yun. Survey on the analysis and modeling of visual kinship: A decade in the making. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4432–4453, 2021. DOI: 10.1109/TPAMI.2021.3063078.

[37] Robinson, Joseph P., Shao, Ming, Wu, Yue, and Fu, Yun. Families in the wild (FIW): Large-scale kinship image database and benchmarks. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 242–246, New York, NY, USA, 2016. ACM. DOI: 10.1145/2964284.2967219.

[38] Robinson, Joseph P, Shao, Ming, Zhao, Handong, Wu, Yue, Gillis, Timothy, and Fu, Yun. Recognizing families in the wild (RFIW) data challenge workshop in conjunction with ACM MM 2017. In *Proceedings of the 2017 Workshop on Recognizing Families in the Wild*, pages 5–12, 2017. DOI: 10.1145/3134421.3134424.

[39] Robinson, Joseph P., Yin, Yu, Khan, Zaid, Shao, Ming, Xia, Siyu, Stopa, Michael, Timoner, Samson, Turk, Matthew A., Chellappa, Rama, and Fu, Yun. Recognizing families in the wild (RFIW): The 4th edition. In *15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 857–862, 2020. DOI: 10.1109/FG47880.2020.00138.

[40] Schroff, Florian, Kalenichenko, Dmitry, and Philbin, James. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. DOI: 10.1109/CVPR.2015.7298682.

[41] Sukhbaatar, Sainbayar, Weston, Jason, Fergus, Rob, et al. End-to-end memory networks. *Advances in Neural Information Processing Systems*, 28:2440–2448, 2015.

[42] Sun, Yi, Chen, Yuheng, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face representation by joint identification-verification. *Advances in Neural Information Processing Systems*, 27:1988–1996, 2014.

[43] Sun, Yi, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014. DOI: 10.1109/CVPR.2014.244.

[44] Tang, Gongbo, Müller, Mathias, Gonzales, Annette Rios, and Sennrich, Rico. Why self-attention? A targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, 2018. DOI: 10.48550/arXiv.1808.08946.

[45] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[46] Wang, Fei, Jiang, Mengqing, Qian, Chen, Yang, Shuo, Li, Cheng, Zhang, Honggang, Wang, Xiaogang, and Tang, Xiaoou. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017. DOI: 10.1109/CVPR.2017.683.

[47] Wang, Feng, Xiang, Xiang, Cheng, Jian, and Yuille, Alan Loddon. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1041–1049. ACM, 2017. DOI: 10.1145/3123266.3123359.

[48] Wang, Hao, Wang, Yitong, Zhou, Zheng, Ji, Xing, Gong, Dihong, Zhou, Jingchao, Li, Zhifeng, and Liu, Wei. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. DOI: 10.1109/CVPR.2018.00552.

[49] Wang, S., Ding, Z., and Fu, Y. Cross-generation kinship verification with sparse discriminative metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. DOI: 10.1109/TPAMI.2018.2861871.

[50] Wang, Shuyang, Robinson, Joseph P, and Fu, Yun. Kinship verification on families in the wild with marginalized denoising metric learning. In *12th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 216–221, 2017. DOI: 10.1109/FG.2017.35.

[51] Wen, Yandong, Zhang, Kaipeng, Li, Zhifeng, and Qiao, Yu. A discriminative feature learning approach for deep face recognition. In *PRoceedings of the European Conference on Computer Vision*, pages 499–515. Springer, 2016. DOI: 10.1007/978-3-319-46478-7_31.

[52] Woo, Sanghyun, Park, Jongchan, Lee, Joon-Young, and So Kweon, In. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. DOI: 10.1007/978-3-030-01234-2_1.

[53] Wu, Xiaoting, Feng, Xiaoyi, Cao, Xiaochun, Xu, Xin, Hu, Dewen, López, Miguel Bordallo, and Liu, Li. Facial kinship verification: A comprehensive review and outlook. *International Journal of Computer Vision*, pages 1–32, 2022. DOI: 10.1007/s11263-022-01605-9.

[54] Yan, Haibin and Hu, Junlin. Video-based kinship verification using distance metric learning. *Pattern Recognition*, 75:15–24, 2018. DOI: 10.1016/j.patcog.2017.03.001.

[55] Yang, Zichao, Yang, Diyi, Dyer, Chris, He, Xiaodong, Smola, Alex, and Hovy, Eduard. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[56] Yu, Jun, Li, Mengyan, Hao, Xinlong, and Xie, Guochen. Deep fusion siamese network for automatic kinship verification. In *15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 892–899. IEEE, 2020. DOI: 10.1109/FG47880.2020.00127.

[57] Zhang, Han, Goodfellow, Ian, Metaxas, Dimitris, and Odena, Augustus. Self-attention generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*, pages 7354–7363, 2019. DOI: 10.48550/arXiv.1805.08318.

[58] Zhao, Ting and Wu, Xiangqian. Pyramid feature attention network for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3085–3094, 2019. DOI: 10.1109/CVPR.2019.00320.

Received 12th March 2022

Acta Cybernetica **26** (2023) 243–266.

Refined Fuzzy Profile Matching*

Gábor Rácz[ab], Attila Sali[ac], and Klaus-Dieter Schewe[d]

Abstract

A profile describes a set of properties, e.g. a set of skills a person may have or a set of skills required for a particular job. Profile matching aims to determine how well a given profile fits to a requested profile and vice versa. Fuzzyness is naturally attached to this problem. The filter-based matching theory uses filters in lattices to represent profiles, and matching values in the interval [0,1], so the lattice order refers to subsumption between the concepts in a profile. In this article the lattice is extended by additional information in form of weighted extra edges that represent partial quantifiable relationships between these concepts. This gives rise to fuzzy filters, which permit a refinement of profile matching. Another way to introduce fuzzyness is to treat profiles as fuzzy sets. In the present paper we combine these two aproaches. Extra edges may introduce directed cycles in the directed graph of the ontology, and the structure of a lattice is lost. We provide a construction grounded in formal concept analysis to extend the original lattice and remove the cycles such that matching values determined over the extended lattice are exactly those resulting from the use of fuzzy filters in case of crisp profiles. For fuzzy profiles we show how to modify the weighting construction while eliminating the directed cycles but still regaining the matching values. We also give sharp estimates for the growth of the number of vertices in this construction.

Keywords: lattice, filter, matching measure, fuzzy sets, fuzzy filter, lattice enrichment, formal concept analysis

*Research of the second author was partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–116769 and SNN-135643. This work was also supported by the BME-Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC) and by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Artificial Intelligence National Laboratory of Hungary.

[a]Alfréd Rényi Institute of Mathematics, Budapest, Hungary

[b]E-mail: gabee33@gmail.com, ORCID: 0000-0002-0733-1350

[c]E-mail: sali.attila@renyi.hu, ORCID: 0000-0002-4837-6360

[d]Zhejiang University, Hangzhou, Zhejiang, China, E-mail: kdschewe@acm.org, ORCID: 0000-0002-8309-1803

DOI: 10.14232/actacyb.277380

1 Introduction

A profile describes a set of properties, and profile matching is concerned with the problem of determining how well a given profile fits to a requested one. Profile matching appears in many application areas such as matching applicants for jobs to job requirements, matching system configurations to requirements specifications, matching team players to game strategies in sport, etc.

A simple approach to profile matching considers profiles as sets of unrelated items, which leads to measuring the similarity or distance of sets. Several ways of definition of distances of sets were introduced such as Jaccard or Sørensen-Dice measures [17], which turned out to be useful in ecological applications. However, skills or properties included in profiles are usually not totally unrelated items, dependencies between them exist and need to be taken into account. For example, in the human resources area several taxonomies for skills, competences and education such as DISCO [6], ISCED [13] and ISCO [14] have been set up. These taxonomies organize the individual properties into a lattice structure. Popov and Jebelean [26] proposed to define an asymmetric matching measure on the basis of filters in such lattices. They represented a profile P with the lattice filter generated by P on the basis that having a specialized skill imples the having a more general skill like knowledge of Java assumes knowledge of Object Oriented Programming as in Figure 4.

Besides such subsumption relationships captured by the lattice order other "horizontal" relationships exist as well. For instance, a job applicant may have some other skills with certain probabilities or of some (not complete) proficiency level, e.g. we may reasonably assume that knowledge of Java implies knowledge of NetBeans up to a grade of 0.7 (or with probability 0.7). This kind of dependencies are exploited in [27]. The idea is that a given profile is considered better than another one for a given requested profile, if they match equally using the filter-based measure, but the first one has more items implied partially that match the requested profile. In this way we get a refinement of the filter-based matchings using the maximum weight of a path from the profile's nodes to a vertex x. This process results in a set of nodes with grades in [0,1], which can be interpreted as a fuzzy set. Actually, it turns out to be a fuzzy filter [12, 18].

However, the introduction of extra edges may give rise to directed cycles, and the elegance of the uniform filter-based matchings is destroyed. Therefore, we raised the question in [28], if the extra edges can be used to modify the original lattice in such a way that instead of using fuzzy filters ordinary filters in the modified lattice can be exploited, which means that the refinement can be re-interpreted in the context of the filter-based matching theory. The answer to this problem is positive, as we explore in this article.

1.1 Our Contribution

In this article we develop an enriched theory of profile matching centered around the idea from [27] using weighted extra edges in addition to edges defined by the

order in a lattice to capture partial relationships between concepts in a profile. How matching measures can be extended has been shown in our previous work [27].

We now provide a construction that gets rid of directed cycles caused by the extra edges. In doing so we show that all matching results that can be obtained by exploiting extra edges can also be obtained from an extended lattice without such extra edges. That is, the theory of profile matching remains within the filter-based approach that we developed in [21], which underlines the power and universality of this theory. In particular, we emphasize how to obtain the lattices underlying the matching theory from knowledge bases that define concepts used in given and requested profiles, and accordingly we call the lattices also *ontology lattices*. These knowledge bases are grounded in description logics, so the lattice extensions provide also feedback for fine-tuning the knowledge representation, whereas weighted extra-edges are not supported in the knowledge bases. In [21] it was also shown that under mild plausibility constraints on human-defined matchings appropriate weights can be defined such that the filter-based matchings preserve the human-defined rankings, which further enables linear optimization to synchronize matchings with human expertise. These results on learning matchings from human expertise can now be carried over to the refined matching theory.

The extension is done by extending the original ontology lattice by new nodes and weighting of the nodes. The result is a directed acyclic graph, whose structure reflects the different possible path lengths between nodes of the ontology lattice. A directed acyclic graph naturally represents a poset, although not a lattice in general. In order to gain back the lattice structure formal concept analysis is used.

The concept of offers and applications from [27, 28] is extended to fuzzy sets. That is we interpret such formulations as "knowledge of skill X is an advantage" by giving a membership value to skill X in the offer a number from $(0, 1)$, measuring the importance of X. Similarly, applications are also considered as fuzzy sets where the membership values signify the proficiency of the applicant in the given skill.

While the extension of given profiles is natural, e.g. for job applications the consideration of skills derived from extra edges appears natural, as employers may benefit from these skills, it is not so clear whether the requested profiles should be extended as well. On one hand, profiles should be handled uniformly, as they could represent both given and requested profiles. On the other hand, if requested profiles, e.g. requirements in job offers, are also extended, then it may happen that a high matching score may result only from derived skills, not from the ones originally required, which may be considered as being misleading and disadvantageous. In the present paper we discuss both scenarios, the latter one being treated by applying different weighting functions for given and requested profiles.

Note the conceptual difference between horizontal connections represented by extra edges and the membership values of skills in offers and applications. The extra edges belong to the taxonomy used and are determined by the domain experts, while fuzzy values are determined by the firms and individuals who apply the matching measure to rank applications for offers.

1.2 Organization of the Article

The remainder of this article is organized as follows. In Section 2 we provide a brief introduction of the fundamentals of filter-based profile matching as developed in our previous research (summarized in [21]), and then extend the approach by using extra edges and fuzzy filters. Section 3 is then dedicated to the construction of the lattice enlargement using formal concept analysis and the proof that matching values using extra edges can be equivalently obtained by ordinary matching values on the extended lattice. We also give node weightings that preserve the weights of fuzzy filters assuming that requested profiles are also extended. Section 4 contains the analysis for the case that requested profiles are not extended. Section 3.3 discusses related extremal problems concerning how the size of the constructed enlargement relates to the size of the original lattice. It is included for the sake of completeness, the proofs of the statements can be found in [28]. Finally, in Section 5 we discuss related work, and in Section 6 we conclude the article with a brief summary.

2 Profile Matching Based on Lattices and Filters

In this section we briefly present the definitions underlying the matching theory from [21] and its refinement from [27], as well as notations of fuzzy set theory used. Matching theory is based on lattice $\mathcal{L}$, and a profile is represented by a filter $\mathcal{F}$ in the lattice. A *matching measure* is a function defined on pairs of filters. If μ is such a matching measure and $\mathcal{F}$, $\mathcal{G}$ are filters, then $\mu(\mathcal{F},\mathcal{G})$ will be a real number in the interval $[0,1]$, which is called a *matching value*. Matching measures in general exploit weights assigned to concepts in the lattice $\mathcal{L}$.

Let $\mathcal{L}(S,\leq)$ be a lattice. Informally, for $A,B \in S$ we have $A \leq B$, if the property A subsumes property B, e.g. for skills this means that a person with skill A will also have skill B. A *filter* is a non-empty subset $\mathcal{F} \subseteq S$, such that for all C, C' with $C \leq C'$ whenever $C \in \mathcal{F}$ holds, then also $C' \in \mathcal{F}$ holds.

Let $\mathbb{F} \subseteq \mathcal{P}(S)$ denote the set of filters. A *weighting function* on S is a function $w\colon \mathcal{P}(S) \to [0,1]$ satisfying (1) $w(S) = 1$, and (2) $w(\bigcup_{i\in I} A_i) = \sum_{i\in I} w(A_i)$ for pairwise disjoint A_i $(i \in I)$.

Definition 1. A *matching measure* is a function $\mu\colon \mathbb{F} \times \mathbb{F} \to [0,1]$ such that $\mu(\mathcal{F}_1,\mathcal{F}_2) = w(\mathcal{F}_1 \cap \mathcal{F}_2)/w(\mathcal{F}_2)$ holds for some weighting function w on $\mathcal{L}$.

The matching measure μ_{pj} defined in [26] uses simply cardinalities:

$$\mu_{pj}(\mathcal{F}_1,\mathcal{F}_2) = \#(\mathcal{F}_1 \cap \mathcal{F}_2)/\#\mathcal{F}_2$$

Thus, it is defined by the weighting function w on S with $w(A) = \#A/\#\mathcal{L}$, i.e. all properties have equal weights. From Section 3 onwards we will tacitly assume that properties have equal weight. This will simplify our presentation, and the extension of our theory to matching measures with general weighting functions is straightforward.

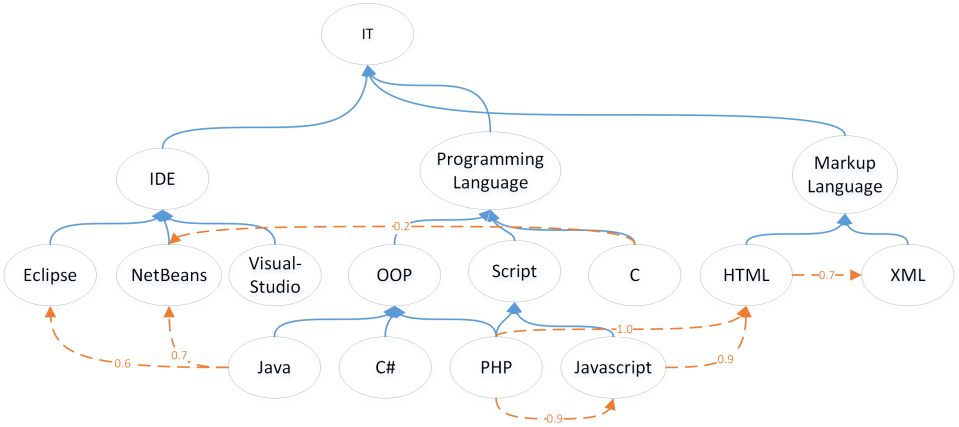


Figure 1: Fragment of a graph with lattice edges (solid) and extra edges (dashed) and assignment of degrees.

Let $\mathbf{F}(S)$ be the collection of fuzzy sets over S. For an $X \in \mathbf{F}(S)$ and $s \in S$ let $\mu_X(s)$ denote the membership value of s in X. We also write $X = \{x_1 : \gamma_1, x_2 : \gamma_2, \ldots, x_n : \gamma_n\}$ where $\gamma_i = \mu_X(x_i)$. The *support* of fuzzy set $X \in \mathbf{F}(S)$ is $\mathsf{supp}(X) = \{s \in S | \mu_X(s) > 0\}$. For two fuzzy sets F, G of $\mathbf{F}(S)$ let $F \cap G = \{s : \gamma_s | s \in S \text{ and } \gamma_s = \min\{\mu_F(s), \mu_G(s)\}$, furthermore let $||F|| := \sum_{v:\gamma_v \in F} \gamma_v$, i.e. $||\cdot||$ denotes sigma cardinality and intersection is defined as the *min* t-norm. Note, that other cardinality and intersection functions could be applied in the same way [35, 12]. We assume that $v : \gamma_v \in F$ means $\gamma_v = \mu_F(v) > 0$.

We can extend the lattice with additional information in form of so called extra edges that represent some kind of quantifiable relationship between skills. However, these edges can form cycles in the hierarchy therefore we use directed graphs to handle them instead of the lattice structure [27].

Let $G = (V, E)$ be a directed graph where $V = S$ and $E = E_{lat} \cup E_{ext}$ is a set of lattice edges and extra edges such that for two nodes $v_i, v_j \in V : (v_i, v_j) \in E_{lat}$ iff v_j covers v_i, i.e. $v_i < v_j$ and there exists no v_k such that $v_i < v_k < v_j$. Furthermore, $(v_i, v_j) \in E_{ext}$ iff there is an extra edge between v_i and v_j. Let $w_{edge} : E \to [0, 1]$ be an edge weighting function such that for all $e_{lat} \in E_{lat} : w_{edge}(e_{lat}) = 1$ and for all $e_{ext} \in E_{ext} : w_{edge}(e_{ext}) \in [0, 1]$ that represents the strength of the relationship between start and end node of the edge. See Figure 1 for a fragment of such a graph. Let $p_F(x, v)$ denote the set of directed paths from node x to node v using edges of a subset $F \subseteq E$ of edge set E of G.

Let application A and offer O be fuzzy sets over S and define a matching function of an application A to an offer O using the graph in the following way. First, we define function *ext* to extend the application and the offer with all the skills that are available from them via directed path in G. For an arbitrary fuzzy set of skills

$X \in \mathbf{F}(S)$ and a subset $F \subseteq E$ of edges let

$$ext_F(X) = \{v{:}\,\gamma_v | v \in S \text{ and } \gamma_v = max_{x',p\in p_F(x',v)} length(p) \cdot \mu_X(x')\}, \quad (1)$$

where length of a path $p = (v_1, \ldots, v_n)$ is the product of the edge weights on p, i.e. $length(p) = \prod_{i=1}^{n-1} w_{edge}((v_i, v_{i+1}))$ and if $p_F(x', v) = \emptyset$, then naturally $length(p) = 0$ for $p \in p_F(x', v)$.

Fuzzy filters were introduced in [18]. A fuzzy set Y over S is a fuzzy filter in $\mathcal{L} = (S, \leq)$ if for all $t \in [0, 1]$ the level set $Y_t = \{y \in Y : \mu_Y(y) \geq t\}$ is a filter in $\mathcal{L}$. A crisp version of the following was proven in [28].

Theorem 1. *Let $G = (S, E = E_{lat} \cup E_{ext})$ be a directed graph with edge weights $w_{edge}: E \to [0, 1]$ extending the lattice $\mathcal{L}(S, \leq)$, and let $X \in \mathbf{F}(S)$ be a fuzzy set over S. Then the extension $ext_E(X)$ of X with respect to E is a fuzzy filter in $\mathcal{L}$.*

Proof. Let $s \in ext_E(X)_t$ and $s < s'$ in $\mathcal{L}$. Furthermore, let $x \in S$ and $p \in p_E(x, s)$ where the maximum in (1) is taken. Since $s < s'$ in $\mathcal{L}$, there exists a directed path p' from s to s' using only lattice edges. The concatenation of p and p' is a directed *walk* q in G from x to s' such that $length(p) = length(q)$, because lattice edges have weight 1. Let q' be the the walk from x to s' of fewest edges such that $q' \subseteq q$. Then clearly $q' \in p_E(x, s')$ and $length(q') \geq length(q) = length(p)$. Hence, $\gamma_{s'} \geq \gamma_s$ implying that $s' \in ext_E(X)_t$. □

Example 1. For the graph in Figure 1 take the following fuzzy sets of skills

$$O = \{Java{:}\,1.0, Netbeans{:}\,0.9, XML{:}\,0.5\} \text{ and}$$
$$A = \{Java{:}\,1.0, PHP{:}\,0.9, Eclipse{:}\,0.7\}$$

These generate the following fuzzy filters:

$$ext_E(O) = \{Java{:}\,1.0, Netbeans{:}\,0.9, XML{:}\,0.5, OOP{:}\,1.0, PL{:}\,1.0,$$
$$IT{:}\,1.0, IDE{:}\,0.9, Eclipse{:}\,0.8, ML{:}\,0.5\}$$

and

$$ext_E(A) = \{Java{:}\,1.0, PHP{:}\,0.9, Eclipse{:}\,0.8, OOP{:}\,1.0, PL{:}\,1.0,$$
$$IT{:}\,1.0, Script{:}\,0.9, IDE{:}\,0.8, Netbeans{:}\,0.7,$$
$$Javascript{:}\,0.81, HTML{:}\,0.9, ML{:}\,0.9, XML{:}\,0.63\}$$

This gives rise to the intersection fuzzy filter

$$ext_E(A) \cap ext_E(O) = \{Java{:}\,1.0, OOP{:}\,1.0, PL{:}\,1.0, IT{:}\,1.0, IDE{:}\,0.8,$$
$$Eclipse{:}\,0.8, Netbeans{:}\,0.7, XML{:}\,0.5, ML{:}\,0.5\}$$

Assuming a weighting function w that assigns the same weight to all elements we obtain the matching value $\mu(ext_E(A), ext_E(O)) = \frac{6}{7.1}$.

It perfectly makes sense to use lattice edges to extend applications and offers as lattice edges describe specialization relation between skills. Namely if an applicant

possesses a special skill then he or she must possess the more general skills as well. However extra edges are used in the extension as well to get more selective matching functions that help differentiate applications.

Let us call nodes in $\mathsf{supp}(ext_E(X)) \setminus \mathsf{supp}(ext_{E_{lat}}(X))$ *derived* nodes for a fuzzy set $X \in \mathbf{F}(S)$ of skills. We investigate two approaches or philosophies when extending profiles using the extra edges. The first one is *symmetric*, that is the case when offers and applications are treated in the same way. In this case we use extension function ext_E for both, offers O and applications A. The advantage is that we only have to apply one weighting function and the proof of equivalence of different representations is simpler than that of the other case. There is a disadvantage, though. If offers are also extended with derived skills, then an application may obtain high matching value just having those skills. However, it is not really advantageous for an employer, as required skills are not in the application.

The second approach called the *strict approach* is when offers are only extended with non-derived nodes, that is ext_E is used for applications but $ext_{E_{lat}}$ is used for offers. This is the approach of [27]. The disadvantage of this case is that different weighting functions have to be applied for applications and offers, consequently the proofs of equivalences are more complicated. However, the point of view of employers is better represented in the second way. An application has to have good matching in target skills to score high, and the derived skills can be used to rank applications scoring equally otherwise. Note, that $\mathsf{supp}(ext_{E_{lat}}(X))$ is exactly the set of nodes contained in the lattice filter generated by the support $\mathsf{supp}(X) = \{s \in S \colon \mu_X(s) > 0\}$ in the ontology lattice $(S, \leq)$.

We adapted the profile matching function proposed by Popov et. al. [26] to fuzzy sets in [27]. We use the same function here except the different approaches in extension of offers. So, let the matching value of A to O be

$$match_{sym}(A, O) = \frac{||ext_E(A) \cap ext_E(O)||}{||ext_E(O)||} \tag{2}$$

in case of the symmetric approach, and

$$match(A, O) = \frac{||ext_E(A) \cap ext_{E_{lat}}(O)||}{||ext_{E_{lat}}(O)||} \tag{3}$$

in case of the strict approach.

3 Lattice Enlargement

In this section, we present a graph transformation method to eliminate extra edges from extended lattices preserving symmetric matching values of applications to offers, and then we use formal concept analysis to restore lattice properties in the transformed graphs.

3.1 Extension Graph

Let $G = (V, E)$ be a directed graph with weighting function w_{edge} as defined above and c_{ij} be the length of the longest path from v_i to v_j where $v_i, v_j \in V$ are two nodes. Let $v_{i_1 j}, \dots v_{i_k j}$ be the nodes from where v_j is available via directed path such that $c_{i_1 j} \leq \dots \leq c_{i_k j}$. Let $c^{j_1}, \dots, c^{j_{l_j}}$ denote the different values among $c_{i_1 j}, \dots, c_{i_k j}$, i.e. $c^{j_1} < \dots < c^{j_{l_j}}$.

For all $c^{j_1} \dots c^{j_{l_j}}$, add new nodes $V_j = \{v_{j_1}, \dots, v_{j_{l_j - 1}}\}$ (for simplicity let $v_{j_{l_j}} = v_j$) to V and add new edges of weight one from $v_{j_{l_j}}$ to $v_{j_{l_j - 1}}$, ..., from v_{j_2} to v_{j_1}, and from v_{j_1} to the top to E. The new edges form a directed path from v_j to the top. Let $q_j = (v_{j_{l_j}}, \dots, v_{j_1}, top)$ denote that path. Assign weight $w_{j_k} = c^{j_k} - c^{j_{k-1}}$ to v_{j_k} $(k = 1, \dots, l_j)$ where $c^{j_0} = 0$. Note, that $\sum_{k=1}^{l} w_{j_k} = 1$ as it is a telescopic sum. If the length of the longest path from v_i to v_j was c^{j_k}, then add a new edge of weight one from v_i to v_{j_k}. Finally, remove all extra edges from the graph. Now each edge has weight one, so edge weights can be ignored.

Let $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ denote the modified graph, called *extension graph*, and w_{node} denote the node weighting function defined.

New nodes of V_j and new edges of q_j can be considered as an extension of v_j to a chain because there do not start edges from intermediate nodes to other chains so out-degrees of intermediate nodes are always one. We call v_j the *base node* of the chain. Base nodes of such chains are nodes of $\mathcal{L}$, and G as well.

Let q_j and q_k be two chains with base nodes v_j and v_k, respectively. Then, an edge from q_k to q_j in G' can go

- from v_k to v_j and then it represents a directed path in G from v_k to v_j containing lattice edges only;
- from v_k to an intermediate node v_{j_i} of q_j and then it represents a directed path $p_{v_k v_j}$ of G from v_k to v_j such that $length(p_{v_k v_j}) = \sum_{s=1}^{i} w_{node}(v_{j_s})$.

Note, that lattice edges in G are acyclic so the corresponding edges in G' are acyclic as well, and newly added edges between different chains start from base nodes of chains only. So G' is an acyclic graph.

Figure 2 shows an example of the construction of G'. There is the original graph, called G, on the left. Blue (solid) edges represent lattice edges and orange (dashed) edges with numbers on them represent extra edges and their weights. There is the extension graph, called G', on the right where green edges represent the newly added edges, and numbers in the top right corners of nodes are weights of the nodes.

As it can be seen, for example, node A of G has been transformed into the chain $q_A = (A, A_1, Top)$ since A is available via lattice edges (i.e. via maximum length paths) from $B, C, Bottom$ and it is available from D via the path $p_{DA} = (D, C, A)$ whose length is 0.8 and A is not available from any other nodes. Therefore A_1 got the weight 0.8 and A got the weight 0.2.

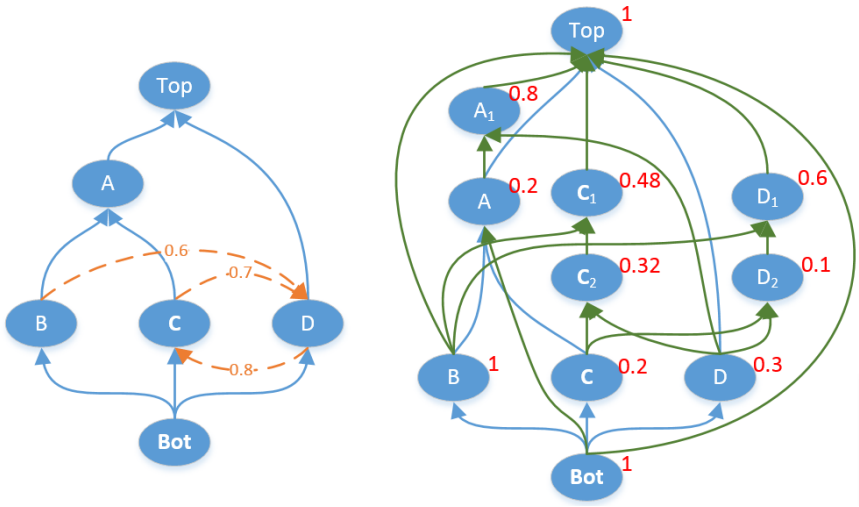


Figure 2: Lattice with extra edges and the generated extension graph

The extension graph defined above makes calculating matching value of *crisp* offers and applications easy. The following was proven in [28], we include the proof here for further use and sake of completeness.

Lemma 1. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph. Let $O \subseteq S$ be an offer and $A \subseteq S$ be an application, that are crisp sets. Then,*

$$match_{sym}(A, O) = \frac{||ext_E(A) \cap ext_E(O)||}{||ext_E(O)||} = \frac{||ext_{E'}(A) \cap ext_{E'}(O)||}{||ext_{E'}(O)||}, \qquad (4)$$

where $ext_{E'}(X)$ denotes the set of vertices of G' that are reachable from some nodes in X via directed paths of G'.

Proof. Let $u \in G'$ and let $q_z = (z_{l_z}, \ldots, z_1, top)$ be the node chain with base node $z \in G$ that contains u, i.e. $z_{l_z} = z$ and $u = z_i$ for some $i \in [1 \ldots l_z]$. First, we will show for an arbitrary $X \subseteq S$ that $u \in ext_{E'}(X)$ iff $z \in ext_E(X)$.

If $u \in ext_{E'}(X)$, then there is a node $a \in X \subseteq V'$ and a directed path $p_{au} = (x_1, \ldots, x_i, x_{i+1}, \ldots x_n)$ from a to u in G' where $x_1 = a$ and $x_n = u$. If $a = z$ then $z \in ext_E(X)$. Otherwise let $x_{i+1} = z_m$ be the first node of p_{au} that is an intermediate node of q_z as well. Then for $j \in [1 \ldots i-1]$: x_j, x_{j+1} are nodes of G, and edges (x_j, x_{j+1}) of p_{au} represent directed paths containing lattice edges only in G. Note that lattice edges form an acyclic subgraph of G. Therefore the concatenations of lattice edge paths $p_{x_j x_{j+1}}$ represented by directed edges (x_j, x_{j+1}) of G' for $j \in [1 \ldots i-1]$ is a path p_{ax_i} in the lattice $\mathcal{L}$ from a to x_i. Now, the edge $(x_i, x_{i+1} = z_m)$ of G' represents a directed path $p_{x_i z}$ from x_i to z in G using some

extra edges. The concatenation of p_{ax_i} and p_{x_iz} is a directed walk from a to z in G, so it contains a directed path p_{az}, that is $z \in ext_E(X)$.

On the other hand, if $z \in ext_E(X)$ with grade γ_z, then there is a node $b \in X$ and a maximal length path p_{bz} from b to z in G such that $length(p_{bz}) = \gamma_z$. In that case, there is an edge from b to z_r in G' for some $r \in [1..l_z]$ such that $\sum_{s=1}^{r} w'_{node}(z_r) = length(p_{bv})$ and also $z_r, z_{r-1}, z_1 \in ext_{E'}(X)$.

Consequently, $ext_{E'}(A) \cap ext_{E'}(O)$ contains fragments of chains generated from base nodes that are available from both A and O in G. Sum of node weights in a fragment equals to the minimum of the lengths of the maximal length paths starting from A or O ending in the base node of the chain. Thus, $||ext_E(A) \cap ext_E(O)|| = ||ext_{E'}(A) \cap ext_{E'}(O)||$ and $||ext_E(O)|| = ||ext_{E'}(O)||$, i.e. equation (4) holds. □

If offers O and applications A are allowed to be fuzzy sets, that is $O, A \in \mathbf{F}(S)$, then the situation is more complicated. As an example consider the lattice and extension graph of Figure 2. If $A = \{D{:}\,0.6\}$ and $O = \{B{:}\,0.9\}$, then $ext_E(A) = \{A{:}\,0.48, C{:}\,0.48, D{:}\,0.6, top{:}\,0.6\}$ and $ext_E(O) = \{A{:}\,0.9, B{:}\,0.9, D{:}\,0.54, C{:}\,0.432, top{:}\,0.9\}$, so $ext_E(A) \cap ext_E(O) = \{A{:}\,0.48, C{:}\,0.432, D{:}\,0.54, top{:}\,0.6\}$. Observe that for any $X \in \mathbf{F}(S)$ we have $\mathsf{supp}(ext_E(X)) = ext_E(\mathsf{supp}(X))$ that would suggest defining $ext'_{E'}(X) = \{v{:}\,\gamma_v | \gamma_v = \max \mu_X(x) w_{node}(v)$ for $x \in \mathsf{supp}(X)$ and $\exists$ directed path from x to v in $G'\}$. However, this definition would give

$$ext'_{E'}(A) = \{A_1{:}\,0.48, C_2{:}\,0.192, C_1{:}\,0.288, D{:}\,0.18, D_2{:}\,0.06, D_1{:}\,0.36, top{:}\,0.6\}$$

and

$$ext'_{E'}(O) = \{A{:}\,0.18, A_1{:}\,0.72, B{:}\,0.9, C_1{:}\,0.432, D_1{:}\,0.54, top{:}\,0.9\}$$

resulting in

$$ext'_{E'}(A) \cap ext'_{E'}(O) = \{A_1{:}\,0.48, C_1{:}\,0.288, D_1{:}\,0.36, top{:}\,0.6\}.$$

Thus, $||ext'_{E'}(A) \cap ext'_{E'}(O)|| \neq ||ext_E(A) \cap ext_E(O)||$.

In order to resolve this problem we charge the contributions of nodes of each chain to the chain's top node as follows. For $x, v \in S$ define $t(x, v) = \sum_{i=1}^{m} w_{node}(v_i)$ where (x, v_m) is the edge of the extension graph G' from x to the chain q_v. If no such edge exists then $t(x, v)$ is defined to be 0. Note, that values $t(x, v)$ can be calculated as a preprocessing step for every pair $x, v \in S$, since they do not depend on particular profiles. Let $X \in \mathbf{F}(S)$ and $x \in \mathsf{supp}(X)$, furthermore let $ext^f_{E'}(X) = \{v_1{:}\,\gamma_{v_1} | v \in S$ and $\gamma_{v_1} = \max_{x \in \mathsf{supp}(X)} \mu_X(x) t(x, v)\}$. Considering the previous example of $A = \{D{:}\,0.6\}$ and $O = \{B{:}\,0.9\}$, we obtain $ext^f_{E'}(A) = \{A_1{:}\,0.48, C_1{:}\,0.48, D_1{:}\,0.6, top{:}\,0.6\}$ and $ext^f_{E'}(O) = \{A_1{:}\,0.9, B_1{:}\,0.9, D_1{:}\,0.54, C_1{:}\,0.432, top{:}\,0.9\}$. Note that $B_1 = B$ and $top_1 = top$ as their chains contain one element, respectively.

Theorem 2. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph. Let $O \in \mathbf{F}(S)$ be an offer and $A \in \mathbf{F}(S)$ be an application . Then*

$$match_{sym}(A, O) = \frac{||ext_E(A) \cap ext_E(O)||}{||ext_E(O)||} = \frac{||ext^f_{E'}(A) \cap ext^f_{E'}(O)||}{||ext^f_{E'}(O)||}. \qquad (5)$$

Proof. There is a directed edge (x, v_m) from a node $x \in S$ to $v_m \in V'$ iff there exists a directed path from x to v in G by the construction of the extension graph. Furthermore, $\sum_{i=1}^{m} w_{node}(v_i)$ the length of the longest path from x to v in G. Thus, for any $X \in \mathbf{F}(S)$ we have $v{:}\,\gamma_v \in ext_E(X) \iff v_1{:}\,\gamma_v \in ext^f_{E'}(X)$.This together with $v \in \mathsf{supp}(ext_E(A) \cap ext_E(O)) \iff v \in \mathsf{supp}(ext_E(A)) \cap \mathsf{supp}(ext_E(O)) \iff v \in \mathsf{supp}(ext^f_{E'}(A)) \cap \mathsf{supp}(ext^f_{E'}(O)) \iff v \in \mathsf{supp}(ext^f_{E'}(A) \cap ext^f_{E'}(O))$ completes the proof. □

Note, that G' is acyclic by its construction but does not necessarily define a lattice. There is a natural way to define a lattice, namely a concept lattice from G' in which matching values of *crisp* applications to *crisp* offers are preserved.

3.2 Concept Lattices

First, we define a formal context and formal concepts based on G'. Let (V_1', V_2', T') be a formal *context*, where $V_1' = V_2' = V'$ and $(v_i, v_j) \in T'$ iff v_j is available from v_i via directed path supposing that the relation is reflexive. Consider the element of V_1' as start points and the element of V_2' as end points of directed paths in G'. Let $I \subseteq V_1'$ and $J \subseteq V_2'$ and let us define their dual sets I^{D_s} and J^{D_e} as follows:

$$I^{D_s} = \{b \in V_2' \mid (a, b) \in T' \text{ for all } a \in I\}$$
$$J^{D_e} = \{a \in V_1' \mid (a, b) \in T' \text{ for all } b \in J\}$$

A *concept* of the context (V_1', V_2', T') is a pair $\langle I, J\rangle$ such that $I \subseteq V_1'$, $J \subseteq V_2'$ and $I^{D_s} = J$, $J^{D_e} = I$. I is called an *extent* of $\langle I, J\rangle$, and J is called an *intent* of $\langle I, J\rangle$.

Table 1: Formal context (V_1', V_2', T')

	Bot	B	C	C1	C2	D	D1	D2	A	A1	Top
Bot	X	X	X	X	X	X	X	X	X	X	X
B		X		X			X		X	X	X
C			X	X	X		X	X	X	X	X
C1				X							X
C2				X	X						X
D				X	X	X	X	X		X	X
D1							X				X
D2							X	X			X
A									X	X	X
A1										X	X
Top											X

Table 1 shows the formal context (V_1', V_2', T') that was generated based on graph G' of Figure 2. Labels of rows and columns represent the elements of V_1' and the

elements of V_2', respectively. There is an X in row i column j if $(i,j) \in T'$, i.e. j is available from i via directed path in G'.

Lemma 2. *If G' is an acyclic graph, then*

1. *For every concept $\langle I, J\rangle$ of the context (V_1', V_2', T'): $I \cap J = \{v\}$ for some $v \in V'$ or $I \cap J = \emptyset$*
2. *For every $v \in V'$: there is a concept $\langle I_v, J_v\rangle$ in the context (V_1', V_2', T') such that $I_v \cap J_v = \{v\}$.*

Proof.

1. Indirectly, suppose that for a concept $\langle I, J\rangle$ of (V_1', V_2', T') and for two different nodes $u, v \in V'$: $u, v \in (I \cap J)$ holds. In this case $(u, v) \in T'$ and $(v, u) \in T'$ hold as well. It would mean that there is a cycle in G' which is a contradiction as G' is acyclic.
2. For a node $v \in V'$ let $J_v = \{v\}^{D_s}$ be the set of all nodes that are available from v via directed path (including v itself). Let $I_v = J_v^{D_e}$, then $v \in I_v$. If $I_v = \{v\}$, then $\langle I_v, J_v\rangle$ is the concept we are looking for.
   Otherwise, suppose that for a node u such that $u \neq v$: $u \in I_v = J_v^{D_e} = (\{v\}^{D_s})^{D_e}$. That means $(u, v) \in T'$, i.e. v is available from u. As T' is a transitive relation $\{v\}^{D_s} \subseteq \{u, v\}^{D_s}$. However $\{u, v\}^{D_s} \subseteq \{v\}^{D_s}$ because $\{u, v\}^{D_s}$ cannot contain such node that is not available from all nodes of $\{u, v\}$. Following this construction we can get that if $J_v^{D_e} = I_v = \{u_1, \ldots, u_i, v\}$, then $I_v^{D_s} = \{u_1, \ldots, u_i, v\}^{D_s} = \{v\}^{D_s} = J_v$. Therefore $\langle \{u_1, \ldots, u_i, v\}, \{v\}^{D_s}\rangle$ is a concept such that $\{u_1, \ldots, u_i, v\} \cap \{v\}^{D_s} = \{v\}$.

□

Let $\mathcal{B}(V_1', V_2', T')$ be the set of all formal concepts in the context, and $\leq$ be a subconcept-superconcept order over the concepts such that for any $\langle A_1, B_1\rangle$, $\langle A_2, B_2\rangle \in \mathcal{B}(V_1', V_2', T')$: $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$, iff $A_1 \subseteq A_2$ (or, iff $B_2 \subseteq B_1$). $(\mathcal{B}(V_1', V_2', T'), \leq)$ is called *concept lattice* [10] and let $cl((\mathcal{L}, E_{ext}))$ denote the concept lattice obtained from the extension graph $ext(\mathcal{L}, E_{ext})$.

Figure 3 [1] shows concept lattice of the context (V_1', V_2', T') from Table 1. Concepts $\langle I_v, J_v\rangle$ where $I_v \cap J_v = \{v\}$ are labeled with v. For example, $\langle I_{C_2}, J_{C_2}\rangle = \langle \{Bot, C, C_2, D\}, \{C_2, C_1, Top\}\rangle$. But, concepts $\langle I, J\rangle$ such that $I \cap J = \emptyset$ are unlabeled like the $\langle \{Bot, B, C\}, \{A, A_1, C_1, D_1, Top\}\rangle$ parent of concepts B and C.

Another, larger example of concept lattice is shown on Figure 4 obtained from the ontology with added extra edges from [27] shown on Figure 1.

It is worth mentioning that the concept lattice $cl((\mathcal{L}, E_{ext}))$ generated from ontology $\mathcal{L}$ endowed with extra edges E_{ext} coincides with the Dedekind-McNeille completion [8] of the poset obtained as transitive closure of acyclic directed graph

[1] The concept lattices were generated using the Concept Explorer tool. Web page: http://conexp.sourceforge.net/

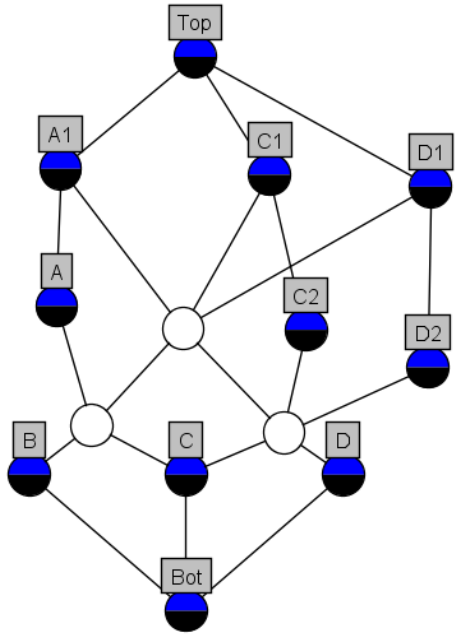


Figure 3: Concept lattice of context (V_1', V_2', T')

$ext(\mathcal{L}, E_{ext})$. Indeed, the collection of upper bounds of a subset S of elements of the poset is exactly the collection of the vertices reachable from the vertices of S via directed paths in the directed graph. We use the concept lattice formulation for two reasons. First, a direct construction is obtained skipping the step of constructing the poset from the directed graph $ext(\mathcal{L}, E_{ext})$. Second, the concept lattice structure allows us to define node weights properly.

A crisp offer $O = \{o_1, \ldots, o_k\} \subseteq S = V \subseteq V'$ generates a filter $F_O \subseteq \mathcal{B}(V_1', V_2', T')$ in the concept lattice such that $F_O = \{\langle I, J\rangle \mid \exists \langle I_o, J_o\rangle \leq \langle I, J\rangle$ such that $I_o \cap J_o = \{o\}$ for some $o \in O\}$. Similarly, a crisp application A generates a filter F_A in the concept lattice.

Let $w_{con}: \mathcal{B}(V_1', V_2', T') \to [0, 1]$ be a concept weighting function such that for a concept $\langle I, J\rangle$ of $\mathcal{B}(V_1', V_2', T')$:

$$w_{con}(\langle I, J\rangle) = \begin{cases} w_{node} & \text{if } I \cap J = \{v\} \text{ for some } v \in V', \\ 0 & \text{otherwise.} \end{cases}$$

Let w_{fil} be a filter weighting function such that for a filter $F \in \mathcal{P}(\mathcal{B}(V_1', V_2', T'))$: $w_{fil}(F) = \sum_{\langle I,J\rangle \in F} w_{con}(\langle I, J\rangle)$.

The following was proven in [28].

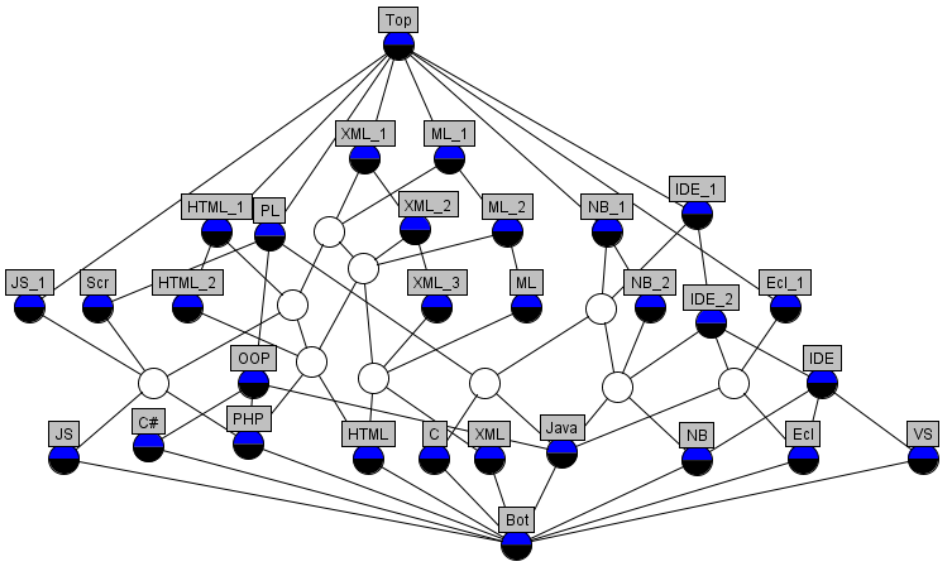


Figure 4: The concept lattice corresponding to the ontology lattice with added extra edges of Figure 1.

Theorem 3. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges and $cl((\mathcal{L}, E_{ext})) = (\mathcal{B}(V_1', V_2', T'), \leq)$ be the concept lattice constructed from G and w_{fil} be the filter weighting function. Let $O \subseteq S$ be an offer and $A \subseteq S$ be an application. Then,*

$$match_{sym}(A, O) = \frac{w_{fil}(F_A \cap F_O)}{w_{fil}(F_O)}. \tag{6}$$

The case of fuzzy offers and applications has the same complication as was with the extension graph. Similarly, we can salvage by charging the contributions of named concepts to the "top one with the same name". That is, we define $t(x, v) = \sum_{\langle I_x, J_x\rangle \leq \langle I_{v_i}, J_{v_i}\rangle \leq \langle I_{v_1}, J_{v_1}\rangle} w_{con}(\langle I_{v_i}, J_{v_i}\rangle)$ if $\langle I_x, J_x\rangle \leq \langle I_{v_1}, J_{v_1}\rangle$ and 0 otherwise, for all pairs $x, v \in S$. This again, is a preprocessing step. For $X \in \mathbf{F}(S)$ let $w_X^f(\langle I_{v_1}, J_{v_1}\rangle) = \{\langle I_{v_1}, J_{v_1}\rangle = \max_{x \in \mathsf{supp}(X)} \mu_X(x) t(x, v)\}$ and $w_X^f(\langle I_{v_i}, J_{v_i}\rangle) = 0$ for $i > 1$, as well as $w_X^f(\langle I, J\rangle) = 0$ if $I \cap J = \emptyset$. Furthermore, for the filter F_X of the concept lattice $\mathcal{B}(V_1', V_2', T'))$ generated by $\mathsf{supp}(X)$ let $fuzz_{fil}(F_X) = \{\langle I, J\rangle : w_X^f(\langle I, J\rangle) | \langle I, J\rangle \in F_X\}$ be a fuzzy set. Then the following can be proven along the lines of the proof of Theorem 2.

Theorem 4. *For a given offer $O \in \mathbf{F}(S)$ and application $A \in \mathbf{F}(S)$ we have*

$$match_{sym}(A, O) = \frac{||fuzz_{fil}(F_A) \cap fuzz_{fil}(F_O)||}{||fuzz_{fil}(F_O)||}. \tag{7}$$

3.3 Extremal problems

It is a natural question how the size of the original ontology lattice $\mathcal{L} = (S, \preceq)$ relates to the sizes of the extension graph $ext(\mathcal{L}, E_{ext})$ and the concept lattice $cl((\mathcal{L}, E_{ext}))$ obtained from $ext((\mathcal{L}, E_{ext}))$.

The proofs of the following statements can be found in the conference paper [28]. First, let us consider $ext(\mathcal{L}, E_{ext})$.

Proposition 1. *Let $\mathcal{L} = (S, \preceq)$ be an ontology lattice of $n + 2$ nodes. Then for $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ we have $|V'| \leq n^2 + 2$. Furthermore, this estimate is sharp, that is for every positive integer n there exists ontology $\mathcal{L}_n = (S_n, \preceq)$ and set of extra edges E_{ext} such that $ext(\mathcal{L}_n, E_{ext})$ has $n^2 + 2$ vertices.*

The extremal example is shown on Figure 5.

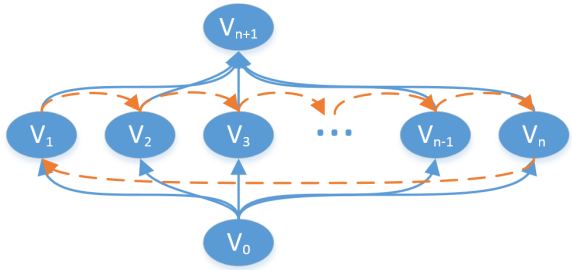


Figure 5: Extremal example

Our next goal is to bound the size of concept lattice $cl((\mathcal{L}, E_{ext}))$. The main question is how many "dummy" vertices are generated, that is concepts $\langle I, J \rangle$ such that $I \cap J = \emptyset$.

Theorem 5. *Let $\mathcal{L} = (S, \preceq)$ be an ontology lattice of $n + 2$ nodes. Then for a set E_{ext} of extra edges $|cl((\mathcal{L}, E_{ext}))| \leq 2^n + n^2 - n + 1$ and this estimate is sharp, that is there exist $\mathcal{L}_n = (S_n, \preceq)$ and and set of extra edges E_{ext} such that $|cl((\mathcal{L}_n, E_{ext}))| = 2^n + n^2 - n + 1$.*

We have the same extremal example shown on Figure 5 as before.

Another interesting question could be how the average or expected size of extension graph and the concept lattice relates to the size of the original ontology lattice. This is the topic of further investigations. The first task is finding a reasonable probability distribution for the extra edges.

4 Strict Approach

As it was mentioned above, extra edges can be used based on different philosophies when extending offers. In this section we investigate how strict matching values of

applications to offers can be preserved in the extension graph and in the concept lattice.

4.1 Preserving Strict Matching for crisp offers and applications

The main problem of preserving strict matching values in the extension graph is if extra edges are used to extend the offer, then extra nodes might appear in the extended offer whose weights are greater then 0. However, to address this problem, special node weighting functions can be defined depending on the offers.

For an offer $O \subset S$ let w^O_{node} be a node weighting function that preserves the weights of the nodes that are available from O via lattice edges in G, and the nodes that were generated from such nodes in G', and it assigns 0 to the other nodes, i.e. for a node $v \in V'$ let

$$w^O_{node}(v) = \begin{cases} w_{node}(v) & \text{if } \exists v_j \in ext_{E_{lat}}(O)\colon v \in V_j, \\ 0 & \text{otherwise.} \end{cases}$$

For $X \subset S$ let $ext^O_{E'}(X) = \{v : w^O_{node}(v) | \exists x \in X$ such that $p_{E'}(x, v) \neq \emptyset\}$ Note, that computing w^O_{node} is a preprocessing step that has to be done once for all offers, and then w^O_{node} can be reused to calculate matching values of applications to the given offer.

With these weighting function a similar result can be shown as in Lemma 1.

Lemma 3. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph, Let $O \subseteq S$ be an offer with w^O_{node} and let $A \subseteq S$ be an application. Then,*

$$match(A, O) = \frac{||ext_E(A) \cap ext_{E_{lat}}(O)||}{||ext_{E_{lat}}(O)||} = \frac{||ext_{E'}(A) \cap ext^O_{E'}(O)||}{||ext_{E'}(O)||} \tag{8}$$

Proof. The proof is analogous to Lemma 1's. However, $ext_{E'}(A) \cap ext_{E'}(O)$ may contain chain fragment $(v_{y_k}, \ldots, v_{y_1})$ of a chain $q_y = \{v_{y_l}, \ldots, v_{y_1}, top\}$ with base node v_y where v_y is only available from O via extra edges in G, i.e. $v_y \in ext_E(O) \setminus ext_{E_{lat}}(O)$. But w^O_{node} assigns 0 to such $v_{y_k}, \ldots, v_{y_1}$ nodes by definition. Therefore $||ext_{E'}(A) \cap ext^O_{E'}(O)|| = \sum_{u \in ext_{E'}(A) \cap ext_{E'}(O)} \min(w_{node}(u), w^O_{node}(u)) = ||ext_E(A) \cap ext_{E_{lat}}(O)||$ and analogously, $||ext_{E_{lat}}(O)|| = ||ext^O_{E'}(O)||$. Thus equation (8) holds as well. □

The same issue appears if we want to preserve strict matching values of crisp applications to crisp offers in the concept lattice as we solved in case of the extension graph, namely extended offer might contain new nodes with weight greater than 0. However, the offer specific weighting functions solve this issue as well.

We extend w^O_{node} for concepts, namely let w^O_{con} be a concept weighting function generated by an offer O such that for a concept $\langle I, J\rangle$:

$$w^O_{con}(\langle I, J\rangle) = \begin{cases} w_{con}(\langle I, J\rangle) & \text{if } I \cap J = \{v\} \text{ such that } \exists v_j \in ext_{E_{lat}}(O)\colon v \in V_j, \\ 0 & \text{otherwise.} \end{cases}$$

Let w^O_{fil} be the filter weighting function based on w^O_{con}, i.e for a filter $F \in \mathcal{P}(\mathcal{B}(V_1', V_2', T'))$: $w^O_{fil}(F) = \sum_{\langle I,J\rangle \in F} w^O_{con}(\langle I, J\rangle)$.

With these weighting functions, we can prove the following theorem similarly to Theorem 3.

Theorem 6. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges and $cl((\mathcal{L}, E_{ext})) = (\mathcal{B}(V_1', V_2', T'), \leq)$ be the concept lattice constructed from G and w_{fil} be the filter weighting function. Let $O \subseteq S$ be an offer with w^O_{con} and w^O_{fil} concept and filter weighting functions, respectively and let $A \subseteq S$ be an application. Then,*

$$match(A, O) = \frac{w^O_{fil}(F_A \cap F_O)}{w^O_{fil}(F_O)} \tag{9}$$

Proof. Analogously to Theorem 3's proof and based on Lemma 1 it is enough to prove that

$$\frac{w^O_{fil}(F_A \cap F_O)}{w^O_{fil}(F_O)} = \frac{||ext_{E'}(A) \cap ext^O_{E'}(O)}{||ext^O_{E'}(O)||}. \tag{10}$$

However, F_A and F_O contain concepts for all nodes of $ext_{E'}(A)$ and $ext_{E'}(O)$ respectively. But w^O_{con} assigns 0 to such $\langle I_v, J_v\rangle$ concepts where $v \in V'$ is not contained in any chain whose base was available from O in G using lattice edges only. Therefore w^O_{fil} sums up the same values as w^O_{fset}, i.e. equation (10) holds as well. □

4.2 Strict matching for fuzzy offers and applications

If offers and applications are allowed to be fuzzy sets, that is $O, A \in \mathbf{F}(S)$, then we are confonted with the same problem as we saw in the symmetric case. Consider the lattice and extension graph of Figure 2. If $O = \{D\colon 0.6\}$ and $A = \{B\colon 0.9\}$, then $ext_{E_{lat}}(O) = \{D\colon 0.6, top\colon 0.6\}$ and $ext_E(A) = \{A\colon 0.9, B\colon 0.9, D\colon 0.54, C\colon 0.432, top\colon 0.9\}$, so $ext_E(A) \cap ext_{E_{lat}}(O) = \{D\colon 0.54, top\colon 0.6\}$. If again we apply definition for the extension graph mechanically we would get $ext'_{E'}(X) = \{v\colon \gamma_v | \gamma_v = \max \mu_X(x) w_{node}(v)$ for $x \in \mathsf{supp}(X)$ and $p_{E'}(x, v) \neq \emptyset\}$ for applications and $ext'_{E'}(X) = \{v\colon \gamma_v | \gamma_v = \max \mu_X(x) w^O_{node}(v)$ for $x \in \mathsf{supp}(X)$ and $p_{E'}(x, v) \neq \emptyset\}$. However, this definition would give

$$ext'_{E'}(O) = \{D\colon 0.18, D_2\colon 0.06, D_1\colon 0.36, top\colon 0.6\}$$

and

$$ext'_{E'}(A) = \{A\colon 0.18, A_1\colon 0.72, B\colon 0.9, C_1\colon 0.432, D_1\colon 0.54, top\colon 0.9\}$$

resulting in

$$ext'_{E'}(A) \cap ext'_{E'}(O) = \{D_1\colon 0.36, top\colon 0.6\}.$$

Thus, $||ext'_{E'}(A) \cap ext'_{E'}(O)|| \neq ||ext_E(A) \cap ext_E(O)||$.

To avoid this anomaly we again charge the contributions of node weights to the top elements of chains, as in the symmetric case. Recall that for $x, v \in S$

$$t(x,v) = \sum_{i=1}^{m} w_{node}(v_i), \tag{11}$$

where (x, v_m) is the edge of the extension graph G' from x to the chain q_v. If no such edge exists then $t(x,v)$ is defined to be 0. Also for $X \in \mathbf{F}(S)$, $ext^f_{E'}(X) = \{v_1 : \gamma_{v_1} | v \in S \text{ and } \gamma_{v_1} = \max_{x \in \mathsf{supp}(X)} \mu_X(x) t(x,v)\}$ was introduced. Now, let $x, v \in S$. Define $t^O(x,v)$ by replacing $w_{node}(v_i)$ by $w^O_{node}(v_i)$ in (11). Furthermore for $X \in \mathbf{F}(S)$, let $ext^{fO}_{E'}(X) = \{v_1 : \gamma_{v_1} | v \in S \text{ and } \gamma_{v_1} = \max_{x \in \mathsf{supp}(X)} \mu_X(x)$ $t^O(x,v)\}$. The proof of the following theorem is straightforward analogue of that of Theorem 2

Theorem 7. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph. Let $O \in \mathbf{F}(S)$ be an offer and $A \in \mathbf{F}(S)$ be an application . Then*

$$match(A,O) = \frac{||ext_E(A) \cap ext_{E_{lat}}(O)||}{||ext_{E_{lat}}(O)||} = \frac{||ext^f_{E'}(A) \cap ext^{fO} f_{E'}(O)||}{||ext^{fO}_{E'}(O)||}. \tag{12}$$

5 Related Work

The aim of profile matching is to find the most fitting candidates to given profiles. Due to its various applications areas, it has become a widely investigated topic recently. Profiles can be represented as sets of elements and then numerous set similarity measures [3], such as Jaccard or Sørensen-Dice, are applicable to compute matching values.

There exist methods assuming that elements of profiles are organized into a hierarchy or ontology. For example, Lau and Sure [16] proposed an ontology-based skill management system for eliciting employee skills and searching for experts within an insurance company. Ragone et al. [29] investigated peer-to-peer e-market place of used cars and presented a fuzzy extension of Datalog to match sellers and buyers based on required and offered properties of cars. Di Noia et al. [5] placed matchmaking on a consistent theoretical foundation using description logic. They defined matchmaking as information retrieval task where demands and supplies are expressed using the same semi-structured data in form of advertisement and task results are ranked lists of those supplies best fulfilling the demands.

Guedj [11] claims that applying semantic matching technologies has the problem that requesting to the user to weighing the skills is a barrier to an usability and an efficiency of such methods on the user point of view and propses a first approach to solve this problem. Tinelli et.al [33] combine the representation power of a logical language with the information processing efficiency of a DBMS and implement it in the platform I.M.P.A.K.T. Shen et.al. [32] use AI to jointly model job description, candidate resume and interview assessment.Yan et.al. [36] realize that interviewers

and job seekers have preferences and propose to learn job-resume matching methods with the hidden preference information incorporated. Pitukhin et.al. [25] take "one sided" question: they present methods to gather and rank job offers from the point of view of the applicant, starting from the assumption that there are many offers that could not be properly assessed by hand.

With respect to foundations of a profile matching theory the first promising attempt to take hierarchical dependencies into account was done by Popov and Jebelean [26], which defines the initial filter-based measure. However, weights are not used, only cardinalities, which correspond to the special case that all concepts are equally weighted. The matching theory in [21] is inspired by this work, but takes the filter-based approach much further. To our best knowledge no other approach in this direction has been tried, though sophisticated taxonomies in the recruitment domain such as DISCO [6], ISCO [14] and ISCED [13] already exist. Ontologies have also been used in the area of recruiting in connection with profile matching (see [7] for a survey). However, while it is claimed that matching accuracy can be improved [23], the matching approach itself remains restricted to Boolean matching, which basically means to count how many requested skills also appear in a given profile [22].

In [27] an extension to the matching theory has been proposed, which exploits relations between the concepts in a profile that are not covered by the lattice, i.e. the presence of a particular concept in a profile may only partially imply the presence of another concept. Such additional links between the elements of the lattice may be associated with a degree (or probability) and even cycles may be permitted. This leads to an *enriched matching theory* by means of values associated to paths, which enables an interpretation using fuzzy filters [12]. For the probabilistic extension this research exploits probabilistic logic with maximum entropy semantics in [1, 15], for which sophisticated reasoning methods exist [30]. In the meantime this research has been taken further showing that it is possible to compute an extended lattice such that matching measures for profiles in the extended lattice capture exactly the same as the path values [28].

We also assumed a structure among elements of profiles that can be represented by an ontology, which then fulfills lattice properties, so profiles can be represented as filters. However, we extended the ontology lattice with extra edges to capture such relationships that subsumptions cannot express. Then we showed how these edges are usable to refine the ontology.

There are several methodologies to learn ontologies from unstructured texts or semi-structured data [2, 31]. Besides identifying concepts, discovering relationships between the concepts is a crucial part of ontology construction and refinement. Text-To-Onto [20] uses statistical, data mining, and pattern-based approaches over text corpus to extract taxonomic and non-taxonomic relations. In [34], various similarity measures were introduced between semi-structured Wikipedia infoboxes and then SVMs and Markov Logic Networks were used to detect subsumptions between infobox-classes.

We presented a method to refine an ontology based on extra edges that represent some sort of quantifiable relationship between concepts in a profile. These

relationships can be given by domain experts, computed from statistics, or result from data mining techniques. For example, in [37] the authors used association rules and latent semantic indexing over job offers to detect relationships between competencies. In our method we defined profile extensions and weighting functions as well to preserve matching values of profiles computed from edge weights.

Formal concept analysis (FCA) [9] is also used to build and maintain formal ontologies. For example, Cimiano et al. [4] presented a method of automatic acquisition of concept hierarchies from a text corpus based on FCA. In [19], the authors used FCA to revise ontology when new knowledge was added to it. In our method we used FCA to restore lattice properties after added new nodes and edges to it based on extra edges. However as we focused on preserving matching values of profiles during the transformations, we adapted our profile weighting functions to the modified ontology lattice as well.

6 Conclusions

In this article the approach of [28] was extended to fuzzy sets of offers and applications. We refined the matching theory with profiles represented by filters in a lattice. Such a lattice can be obtained from a knowledge base as shown in [24]. The basis for the theory is the definition of weighted matching measures on pairs of such filters. For instance, in the field of human resources profiles correspond to skills sets of job applicants as well as to requirements in job offers. Learning matching weight from human expertise as well as efficient querying have been handled in [21]. We now investigated how ontology lattices can be extended by additional information and used for matching. We defined matching functions to find the most suitable applicant to a job offer, however, our results are applicable in other fields as well.

First, profiles are represented as filters in an ontology lattices, which capture subsumption relations between concepts. Then, we extend such an ontology lattice by additional information in the form of extra edges describing additional quantifiable relations between the concepts. A directed graph is built from the lattice endowed with extra edges to handle directed cycles that the new edges might have introduced, and matching functions are defined based on reachable nodes from the nodes in a profiles.

Two approaches were presented to extend profiles with derived nodes. In the first one, both the given and the requested profiles were extended, as profiles should be handled uniformly. In the second approach, only the given profiles were extended, which helps to distinguish cases, where the given requirements are met directly from those, where the requirements are only met by the combination of several concepts that all contribute partially to the requirements. For instance, in the human resources field the second strict approach may help employers to better differentiate among job applicants.

We presented a method that eliminates directed cycles from the graph. It constructs an extension graph by adding node chains to the original lattice based on directed paths between nodes in the directed graph and node weights got also

modified as part of the construction. The extension graph is a directed acyclic graph and therefore a poset but it is not necessary a lattice. We further exploited formal concept analysis to extend the poset to a concept lattice so that filters of this lattice could be used to calculate matching values. Different node weightings were used to preserve the original matching values in the two approaches. Comparisons of the sizes of the ontology lattice and the generated acyclic directed graph, as well as the concept lattice were also given.

This shows that the matching theory from [21] is rather powerful, as it captures de facto the fuzzy extensions.

References

[1] Beierle, Christoph, Finthammer, Marc, and Kern-Isberner, Gabriele. Relational probabilistic conditionals and their instantiations under maximum entropy semantics for first-order knowledge bases. *Entropy*, 17(2):852–865, 2015. DOI: 10.3390/e17020852.

[2] Buitelaar, Paul, Cimiano, Philipp, and Magnini, Bernardo. Ontology learning from text: An overview. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Evaluation and Applications*, Volume 123 of *Frontiers in Artificial Intelligence and Applications*, pages 3–12. IOS Press, Amsterdam, 2005. URL: https://pub.uni-bielefeld.de/record/2497696.

[3] Choi, Seung-Seok, Cha, Sung-Hyuk, and Tappert, Charles C. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

[4] Cimiano, Philipp, Hotho, Andreas, and Staab, Steffen. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24(1):305–339, 2005. DOI: 10.1613/jair.1648.

[5] Di Noia, Tommaso, Di Sciascio, Eugenio, and Donini, Francesco M. Semantic matchmaking as non-monotonic reasoning: A description logic approach. *Journal of Artificial Intelligence Research*, 29:269–307, 2007. DOI: 10.1613/jair.2153.

[6] European dictionary of skills and competences. URL: http://www.disco-tools.eu.

[7] Falk, Thorsten et al. Semantic-Web-Technologien in der Arbeitsplatzvermittlung. *Informatik Spektrum*, 29(3):201–209, 2006. DOI: 10.1007/s00287-006-0061-4.

[8] Ganter, Bernhard and Kuznetsov, Sergei O. Stepwise construction of the Dedekind-MacNeille completion. In Mugnier, Marie-Laure and Chein, Michel,

editors, *Conceptual Structures: Theory, Tools and Applications*, pages 295–302, Berlin, Heidelberg, 1998. Springer. DOI: 10.1007/BFb0054922.

[9] Ganter, Bernhard, Stumme, Gerd, and Wille, Rudolf. Formal concept analysis: Theory and applications. *Journal of Universal Computer Science*, 10(8):926, 2004. DOI: 10.3217/jucs-010-08.

[10] Ganter, Bernhard and Wille, Rudolf. *Formal Concept Analysis: Mathematical Foundations*. Springer Science & Business Media, 2012.

[11] Guedj, Michaël. Levelized taxonomy approach for the job seeking/recruitment problem. In *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) and 15th International Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pages 448–451, 2016. DOI: 10.1109/CSE-EUC-DCABES.2016.222.

[12] Hájek, Petr. *Mathematics of Fuzzy Logic*. Kluwer Academic Publishers, 1998.

[13] International standard classification of education. URL: http://www.uis.unesco.org/Education/Pages/international-standard-classification-of-education.aspx.

[14] International standard classification of occupations. URL: http://www.ilo.org/public/english/bureau/stat/isco/isco08/, 2008.

[15] Kern-Isberner, Gabriele and Lukasiewicz, Thomas. Combining probabilistic logic programming with the power of maximum entropy. *Artificial Intelligence*, 157(1-2):139–202, 2004. DOI: 10.1016/j.artint.2004.04.003.

[16] Lau, Thorsten and Sure-Vetter, York. Introducing ontology-based skills management at a large insurance company. In *Modellierung (Proceedings)*, pages 123–134, 2002.

[17] Levandowsky, Michael and Winter, David. Distance between sets. *Nature*, 234(5):34–35, 1971. DOI: 10.1038/234034a0.

[18] Liu, Lianzhen and Li, Kaitai. Fuzzy filters of BL-algebras. *Information Sciences*, 173(1):141–154, 2005. DOI: 10.1016/j.ins.2004.07.009.

[19] Looser, Dominic, Ma, Hui, and Schewe, Klaus-Dieter. Using formal concept analysis for ontology maintenance in human resource recruitment. In Ferrarotti, Flavio and Grossmann, Georg, editors, *Proceedings of the Ninth Asia-Pacific Conference on Conceptual Modelling*, Volume 143, pages 61–68. Australian Computer Society, 2013. DOI: 10.5555/2527198.2527204.

[20] Maedche, Alexander and Volz, Raphael. The ontology extraction & maintenance framework Text-To-Onto. In *Proceedings of the Workshop on Integrating Data Mining and Knowledge Management*, pages 1–12, 2001.

[21] Martínez Gil, Jorge, Paoletti, Alejandra Lorena, Rácz, Gábor, Sali, Attila, and Schewe, Klaus-Dieter. Accurate and efficient profile matching in knowledge bases. *Data & Knowledge Engineering*, 117:195–215, 2018. DOI: 10.1016/j.datak.2018.07.010.

[22] Mochol, Malgorzata, Nixon, Lyndon J. B., and Wache, Holger. Improving the recruitment process through ontology-based querying. In Bontas Simperl, Elena Paslaru, Hepp, Martin, and Tempich, Christoph, editors, *Proceedings of the First International Workshop on Applications and Business Aspects of the Semantic Web*, Volume 226 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006. URL: https://ceur-ws.org/Vol-226/paper05.pdf.

[23] Mochol, Malgorzata, Wache, Holger, and Nixon, Lyndon J. B. Improving the accuracy of job search with semantic techniques. In Abramowicz, Witold, editor, *Proceedings of the 10th International Conference on Business Information Systems*, Volume 4439 of *Lecture Notes in Computer Science*, pages 301–313. Springer, 2007. DOI: 10.1007/978-3-540-72035-5_23.

[24] Paoletti, Alejandra Lorena, Martinez-Gil, Jorge, and Schewe, Klaus-Dieter. Extending knowledge-based profile matching in the human resources domain. In Chen, Qiming et al., editors, *Database and Expert Systems Applications (DEXA 2015) Part II*, Volume 9262 of *Lecture Notes in Computer Science*, pages 21–35. Springer, 2015. DOI: 10.1007/978-3-319-22852-5_3.

[25] Pitukhin, E., Astafyeva, M., and Astafyeva, I. Methodology for job advertisements analysis in the labor market in metropolitan cities: The case study of the capital of Russia. In Silhavy, R., editor, *Intelligent Algorithms in Software Engineering*, Volume 1224 of *Advances in Intelligent Systems and Computing*. Springer, 2020. DOI: 10.1007/978-3-030-51965-0_37.

[26] Popov, Nikolaj and Jebelean, Tudor. Semantic matching for job search engines – a logical approach. Technical Report 13-02, Research Institute for Symbolic Computation, JKU Linz, 2013.

[27] Rácz, Gábor, Sali, Attila, and Schewe, Klaus-Dieter. Semantic matching strategies for job recruitment: A comparison of new and known approaches. In Gyssens, Marc and Simari, Guillermo Ricardo, editors, *Foundations of Information and Knowledge Systems (FoIKS 2016)*, Volume 9616 of *Lecture Notes in Computer Science*, pages 149–168. Springer, 2016. DOI: 10.1007/978-3-319-30024-5_9.

[28] Rácz, Gábor, Sali, Attila, and Schewe, Klaus-Dieter. Refining semantic matching for job recruitment: An application of formal concept analysis. In Ferrarotti, Flavio and Woltran, Stefan, editors, *Foundations of Information and Knowledge Systems (FoIKS 2018)*, Volume 10833 of *Lecture Notes in Computer Science*, pages 322–339, 2018. DOI: 10.1007/978-3-319-90050-6_18.

[29] Ragone, Azzurra, Straccia, Umberto, Di Noia, Tommaso, Di Sciascio, Eugenio, and Donini, Francesco M. Fuzzy matchmaking in e-Marketplaces of Peer Entities using Datalog. *Fuzzy Sets and Systems*, 160(2):251–268, 2009. DOI: 10.1016/j.fss.2008.07.002.

[30] Schramm, Manfred and Ertel, Wolfgang. Reasoning with probabilities and maximum entropy: The system PIT and its application in LEXMED. In *Operations Research Proceedings 1999*. Springer, 2000. DOI: 10.1007/978-3-642-58300-1_41.

[31] Shamsfard, Mehrnoush and Barforoush, Ahmad Abdollahzadeh. The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, 18(4):293–316, 2003. DOI: 10.1017/S0269888903000687.

[32] Shen, Dazhong, Zhu, Hengshu, Zhu, Chen, Xu, Tong, Ma, Chao, and Xiong, Hui. A joint learning approach to intelligent job interview assessment. In Lang, Jérôme, editor, *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3542–3548, 2018. DOI: 10.24963/ijcai.2018/492.

[33] Tinelli, Eufemia, Colucci, Simona, Donini, Francesco M., Di Sciascio, Eugenio, and Giannini, Silvia. Embedding semantics in human resources management automation via SQL. *Applied Intelligence*, 46:952–982, 2017. DOI: 10.1007/s10489-016-0868-x.

[34] Wu, Fei and Weld, Daniel S. Automatically refining the Wikipedia infobox ontology. In *Proceedings of the 17th International Conference on World Wide Web*, pages 635–644. ACM, 2008. DOI: 10.1145/1367497.1367583.

[35] Wygralak, Maciej. *Cardinalities of Fuzzy Sets*. Springer, 2003. DOI: 10.1007/978-3-540-36382-8.

[36] Yan, Rui, Le, Ran, Song, Yang, Zhang, Tao, Zhang, Xiangliang, and Zhao, Dongyan. Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 914–922. Association for Computing Machinery New York NY, 2019. DOI: 10.1145/3292500.3330963.

[37] Ziebarth, Sabrina, Malzahn, Nils, and Hoppe, Heinz Ulrich. Using data mining techniques to support the creation of competence ontologies. In *Frontiers in Artificial Intelligence and Applications*, pages 223–230, 2009. DOI: 10.3233/978-1-60750-028-5-223.

Received 11th August 2021

Acta Cybernetica **26** (2023) 267–291.

Verifying Provable Stability Domains for Discrete-Time Systems Using Ellipsoidal State Enclosures

Andreas Rauh[a], Auguste Bourgois[bcd], and Luc Jaulin[de]

Abstract

Stability contractors, based on interval analysis, were introduced in recent work as a tool to verify stability domains for nonlinear dynamic systems. These contractors rely on the property that — in case of provable asymptotic stability — a finitely large domain in a multi-dimensional state space is mapped into its interior after a certain integration time for continuous-time processes or after a certain number of discretization steps in a discrete-time setting. However, a disadvantage of the use of axis-aligned interval boxes in such computations is the omnipresent wrapping effect. As shown in this contribution, the replacement of classical interval representations by ellipsoidal domain enclosures reduces this undesirable effect. It also helps to find suitable ratios for the edge lengths if interval-based domain representations are investigated. Moreover, ellipsoidal domains naturally represent the possible regions of attraction of asymptotically stable equilibrium points that can be analyzed with the help of quadratic Lyapunov functions, for which stability criteria can be cast into linear matrix inequality (LMI) constraints. For that reason, this paper further presents possible interfaces of ellipsoidal enclosure techniques with LMI approaches. This combination aims at the maximization of those domains that can be proven to be stable for a discrete-time range-only localization algorithm in robotics. There, an Extended Kalman Filter (EKF) is applied to a system for which the dynamics are characterized by a discrete-time integrator disturbance model with additive Gaussian noise. In this scenario, the measurement equations correspond to the distances between the object to be localized and beacons with known positions.

Keywords: stability analysis, ellipsoidal enclosures, interval methods, Kalman filtering

[a]Carl von Ossietzky Universität Oldenburg, Department of Computing Science, Group: Distributed Control in Interconnected Systems, D-26111 Oldenburg, Germany, E-mail: Andreas.Rauh@uni-oldenburg.de, ORCID: 0000-0002-1548-6547

[b]FORSSEA, Paris, France

[c]E-mail: Auguste.Bourgois@ensta-bretagne.org, ORCID: 0000-0002-0333-5872

[d]ENSTA Bretagne, Lab-STICC, 29806 Brest, France

[e]E-mail: lucjaulin@gmail.com, ORCID: 0000-0002-0938-0615

DOI: 10.14232/actacyb.293871

1 Introduction

The analysis of stability properties of nonlinear dynamic systems is a crucial aspect for the verification of control and state estimation procedures (i.e., state observers) in many different areas. From a methodological point of view, Lyapunov function techniques can be applied to deal with this task for both discrete-time and continuous-time processes [18, 19]. They are not only applicable to the analysis of predefined control and observer structures but are also widely used during their synthesis. Especially when system models with a linear or quasi-linear structure are considered, there exist a large number of interrelations between Lyapunov function techniques and LMIs. This is basically caused by the fact that stability criteria for linear dynamic systems, which are investigated with the help of quadratic candidates for Lyapunov functions, are equivalent to criteria that can be stated with the help of LMIs.

Based on these fundamental observations, numerous research activities have been performed in recent years which (as an obviously non-exhaustive list) deal with the following aspects:

- transforming stability requirements for linear uncertain system models with polytopic time-invariant and time-varying uncertainty into sets of LMIs [2, 5, 8, 9, 40];
- development of iterative LMI techniques for synthesizing robust output and state feedback controllers for systems which are simultaneously subject to polytopic parameter uncertainty and/or stochastic noise [11, 28, 32, 41];
- verifying invariant sets of nonlinear closed-loop control systems [37];
- implementing gain scheduling controllers for quasi-linear systems with bounded parameter uncertainty [17];
- implementing online gain adaptation schemes for variable-structure, sliding mode controllers as well as backstepping techniques with the aim of chattering reduction [33–35];
- investigation of the dual task of variable-structure state estimation [31];
- finding optimal candidates for Lyapunov functions for nonlinear dynamic systems [22, 43];
- determining the region of attraction of stable operating points and maximizing the provable stability domains for nonlinear processes [7, 12, 15, 25, 44–46].

In parallel to the development of the above-mentioned Lyapunov and LMI techniques, interval methods have been investigated during the last decades [16, 20]. Due to their fundamental property to enclose the solution to some mathematically formulated problem in a guaranteed way, they have many applications in engineering. These cover aspects such as state and parameter estimation [1], uncertainty quantification in robotics applications [21, 24], or simulation of dynamic systems [23].

Moreover, a new technique for enclosing provable stability domains was presented recently in [3,4]. This so-called stability contractor is re-investigated in this paper for analyzing stability properties of a discrete-time EKF [42] that is applied to the task of localizing a robot with the help of range-only measurements. For that purpose, the interval-based implementation of this contractor is compared with a novel ellipsoidal enclosure approach. This approach was recently presented in [29,30] as a tool for nonlinear function evaluation, simulation of dynamic system models, as well as performance analysis of linearization-based stochastic filters (such as the EKF). In [26], it has been extended towards a state estimation procedure which exploits a quasi-linear system structure when determining inner and outer bounds for state enclosures.

This paper is structured as follows: Sec. 2 summarizes the already existing interval-based stability contractor and reviews ellipsoidal enclosure techniques for discrete-time dynamic systems. Both provide the basis for the novel ellipsoidal stability contractor in Sec. 3 which enhances the original interval-based technique due to its capability for often proving larger regions of attraction for stable operating points. A (near to optimal) parameterization of this novel contractor is described in Secs. 3.1 and 3.2 with an illustrating example in Sec. 3.3 and its use for a localization task in robotics in Sec. 3.4. Moreover, a new extension for proving instability of equilibrium points is presented in Sec. 3.5. Finally, Sec. 4 describes an outlook on using ellipsoidal techniques for finding positive invariant domains in the frame of continuous-time processes before conclusions are given in Sec. 5.

2 Preliminaries

In this section, fundamental preliminaries published in previous works of the authors are given. These are the interval-based stability contractor [3,4] as well as (thick) ellipsoidal state enclosure techniques for discrete-time systems. For the latter, we make a distinction between a general formulation [29,30] and a specialized version for quasi-linear system models [26].

2.1 Notation

Throughout this paper, scalar interval variables with the lower and upper bounds $\underline{x}$ and $\overline{x}$, respectively, where $\underline{x} \leq \overline{x}$, are denoted as $[x] = [\underline{x}\ ;\ \overline{x}]$. For the vector-valued case, an interval vector (also called *interval box*) is formed as the Cartesian product of scalar intervals according to the stacked notation

$$[\mathbf{x}] = \begin{bmatrix} [x_1] & \dots & [x_n] \end{bmatrix}^T \quad , \tag{1}$$

where the set of axis-aligned interval boxes in $\mathbb{R}^n$ is denoted as $\mathbb{IR}^n$. For fundamental enclosure properties of interval analysis as well as interval extensions of (vector-valued) functions $\mathbf{f} : \mathbb{R}^m \mapsto \mathbb{R}^n$, the reader is referred to [16,20].

Moreover, according to [29,30], define a thick ellipsoid $(\!(\mathscr{E})\!) = (\!(\mathscr{E})\!)\left(\boldsymbol{\mu}, \boldsymbol{\Gamma}, \left[\underline{\rho}\,;\,\overline{\rho}\right]\right)$, where $0 \leq \underline{\rho} \leq \overline{\rho}$, as a subset of the power set $\mathscr{P}(\mathbb{R}^n)$ so that

$$(\!(\mathscr{E})\!) = \left\{ \mathscr{A} \in \mathscr{P}(\mathbb{R}^n) \;\middle|\; \mathscr{E}^{\mathrm{I}} \subseteq \mathscr{A} \subseteq \mathscr{E}^{\mathrm{O}} \right\} \tag{2}$$

encloses a set $\mathscr{A}$ of interest both from the inside and outside with the inner and outer bounding ellipsoids

$$\begin{aligned} \mathscr{E}^{\mathrm{I}} &= \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; (\mathbf{x}-\boldsymbol{\mu})^T \left(\underline{\rho}\boldsymbol{\Gamma}\right)^{-T} \left(\underline{\rho}\boldsymbol{\Gamma}\right)^{-1} (\mathbf{x}-\boldsymbol{\mu}) \leq 1 \right\} \;, \\ \mathscr{E}^{\mathrm{O}} &= \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; (\mathbf{x}-\boldsymbol{\mu})^T (\overline{\rho}\boldsymbol{\Gamma})^{-T} (\overline{\rho}\boldsymbol{\Gamma})^{-1} (\mathbf{x}-\boldsymbol{\mu}) \leq 1 \right\} \end{aligned} \tag{3}$$

that have surfaces parallel to each other.

Finally, $\|\cdot\|$ represents (an interval extension of) the Euclidean norm of the corresponding vector-valued argument as introduced in [29]; the relations $\mathbf{M} \succ 0$ and $\mathbf{M} \succeq 0$ denote positive and positive semi-definiteness of a real-valued symmetric matrix ($\mathbf{M} \prec 0$ and $\mathbf{M} \preceq 0$, negative (semi-) definiteness, respectively).

2.2 Interval-Based Stability Contractors

Consider an interval box $[\mathbf{x}_0]$ of $\mathbb{R}^n$. According to [4, Def. 1], a stability contractor $\Psi : \mathbb{IR}^n \mapsto \mathbb{IR}^n$ of rate $|\alpha| < 1$ is characterized by the following properties for all boxes $[\mathbf{a}], [\mathbf{b}] \subset [\mathbf{x}_0]$:

1. monotonicity: $[\mathbf{a}] \subset [\mathbf{b}] \Longrightarrow \Psi\left([\mathbf{a}]\right) \subset \Psi\left([\mathbf{b}]\right)$;
2. contractance: $\Psi\left([\mathbf{a}]\right) \subset [\mathbf{a}]$;
3. equilibrium: $\Psi(\mathbf{0}) = \mathbf{0}$;
4. convergence: $\Psi\left([\mathbf{a}]\right) \subset \alpha \cdot [\mathbf{a}] \Longrightarrow \forall k \geq 1, \Psi^k\left([\mathbf{a}]\right) \subset \alpha^k \cdot [\mathbf{a}]$, where $\Psi^k\left([\mathbf{a}]\right)$ denotes the iterated evaluation $\underbrace{\Psi \circ \ldots \circ \Psi}_{k}$, where Ψ^0 is the identity function.

As shown in [4], the existence of such a stability contractor with $\Psi\left([\mathbf{x}_0]\right) \subset [\mathbf{x}_0]$ can serve as a proof of Lyapunov stability of a discrete-time dynamic system

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \;, \quad \mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n \tag{4}$$

with the equilibrium state $\mathbf{x} = \mathbf{0}$, i.e., $\mathbf{0} = \mathbf{f}(\mathbf{0})$ in the complete box of initial conditions $[\mathbf{x}_0] \ni \mathbf{0}$.

Remark 1. Due to the fact that a centered form representation of the interval extension of functions such as the system model (4) often leads to tighter bounds of the resulting state enclosures than a naive interval extension if the domain on which the function is evaluated is sufficiently small (cf. [10]), an evaluation of the

stability contractor in centered form representation was proposed in [4]. Moreover, it should be noted that a one-step evaluation of the state equations (especially for systems with oscillatory but asymptotically stable dynamics), often does not satisfy the contractance property mentioned above. Then, the stability contractor can be applied to a multi-time step evaluation by using a k times iterated centered form representation of (4) on the box of initial conditions.

2.3 Ellipsoidal Enclosures for Discrete-Time Dynamic Systems: General Case

Consider a finite-dimensional discrete-time system model (4), where (as also required for the centered form representation in the previous subsection) $\mathbf{f}$ is assumed to be differentiable. Given a thick ellipsoid representation

$$\left(\!\left(\mathscr{E}\right)\!\right)_k = \left(\!\left(\mathscr{E}\right)\!\right)\left(\boldsymbol{\mu}_k, \boldsymbol{\Gamma}_k, \left[\underline{\rho}_k \; ; \; \overline{\rho}_k\right]\right) \tag{5}$$

at the time instant k, a thick ellipsoid

$$\left(\!\left(\mathscr{E}\right)\!\right)_{k+1} = \left(\!\left(\mathscr{E}\right)\!\right)\left(\boldsymbol{\mu}_{k+1}, \boldsymbol{\Gamma}_{k+1}, \left[\underline{\rho}_{k+1} \; ; \; \overline{\rho}_{k+1}\right]\right) \tag{6}$$

at the instant $k+1$ is defined by the following Theorem 1 such that $\mathscr{E}^{\mathrm{I}}_{k+1}$ is an inner boundary containing certainly reachable states and $\mathscr{E}^{\mathrm{O}}_{k+1}$ is a guaranteed outer enclosure. A graphical representation of this enclosure property is given in Fig. 1. For a proof of the following theorem, the reader is referred to [30].

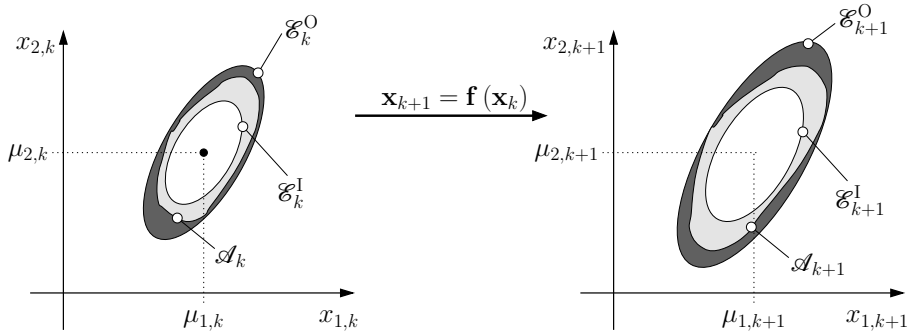


Figure 1: Definition of a thick ellipsoid $\left(\!\left(\mathscr{E}\right)\!\right)_k$ enclosing the domain $\mathscr{A}_k$ and its mapping $\left(\!\left(\mathscr{E}\right)\!\right)_{k+1}$ via the system model (4) that encloses the true solution set $\mathscr{A}_{k+1}$ from the inside and outside.

Theorem 1 ([29,30] Thick ellipsoid enclosures)**.** *Define the state enclosure at the time instant k by the thick ellipsoid $\left(\!\left(\mathscr{E}\right)\!\right)_k$. For a differentiable state equation (4), with*

$$\mathbf{A}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_k}\left(\boldsymbol{\mu}_k\right) \quad \textit{invertible} \;, \tag{7}$$

$\left(\!\left(\mathcal{E}\right)\!\right)_{k+1}$ *according to (6) is a thick ellipsoid enclosure of the solution set* $\mathbf{f}\left(\left(\!\left(\mathcal{E}\right)\!\right)_k\right)$ *with*

$$\boldsymbol{\mu}_{k+1} = \mathbf{f}\left(\boldsymbol{\mu}_k\right) \quad \textit{and} \quad \boldsymbol{\Gamma}_{k+1} = \mathbf{A}_k \cdot \boldsymbol{\Gamma}_k \tag{8}$$

as well as

$$\underline{\rho}_{k+1} = (1 - \rho_{\mathrm{I},k}) \cdot \underline{\rho}_k \quad \textit{and} \quad \overline{\rho}_{k+1} = (1 + \rho_{\mathrm{O},k}) \cdot \overline{\rho}_k \ . \tag{9}$$

Here,

$$\rho_{\mathrm{I},k} = \max_{\|\tilde{\mathbf{x}}_k\| \leq 1} \left\| \tilde{\mathbf{b}}_{\mathrm{I},k}\left(\tilde{\mathbf{x}}_k\right) \right\| \ , \tag{10}$$

$$\tilde{\mathbf{b}}_{\mathrm{I},k}\left(\tilde{\mathbf{x}}_k\right) = \underline{\rho}_k^{-1} \boldsymbol{\Gamma}_k^{-1} \mathbf{A}_k^{-1} \cdot \left(\mathbf{f}\left(\underline{\rho}_k \boldsymbol{\Gamma}_k \tilde{\mathbf{x}}_k + \boldsymbol{\mu}_k \right) - \mathbf{f}\left(\boldsymbol{\mu}_k\right) \right) - \tilde{\mathbf{x}}_k \tag{11}$$

and

$$\rho_{\mathrm{O},k} = \max_{\|\tilde{\mathbf{x}}_k\| \leq 1} \left\| \tilde{\mathbf{b}}_{\mathrm{O},k}\left(\tilde{\mathbf{x}}_k\right) \right\| \ , \tag{12}$$

$$\tilde{\mathbf{b}}_{\mathrm{O},k}\left(\tilde{\mathbf{x}}_k\right) = \overline{\rho}_k^{-1} \boldsymbol{\Gamma}_k^{-1} \mathbf{A}_k^{-1} \cdot \left(\mathbf{f}\left(\overline{\rho}_k \boldsymbol{\Gamma}_k \tilde{\mathbf{x}}_k + \boldsymbol{\mu}_k \right) - \mathbf{f}\left(\boldsymbol{\mu}_k\right) \right) - \tilde{\mathbf{x}}_k \, . \tag{13}$$

2.4 Ellipsoidal Enclosures for Discrete-Time Dynamic Systems: Quasi-Linear Case

As a special case of the general system model (4), consider the quasi-linear system representation

$$\mathbf{x}_{k+1} = \mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right) \cdot \mathbf{x}_k \ , \tag{14}$$

where $\mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right) \in \mathbb{R}^{n \times n}$ is a state- and parameter-dependent system matrix. This matrix can be extracted from the general system formulation (4) either by means of factoring out the state vector in such a way that all matrix entries are finite and non-singular in the operating domain of interest. Alternatively, it can be bounded by means of slope calculus [6] or in analogy to the centered form representation mentioned before by means of an interval extension of the system's Jacobian, see also [26].

Remark 2. To prove asymptotic stability by means of the contractor technique in Sec. 2.2, it is necessary that the matrix $\mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right)$ in (14) does not introduce any further equilibrium point (except for the origin of the state space) in the evaluation domain of interest. This is a direct consequence of the contractance property in Sec. 2.2 which must equally hold for the state equations if the interval-based stability contractor of Sec. 2.2 and the general ellipsoidal evaluation technique of Sec. 2.3 were applied.

The following five-step evaluation procedure for quasi-linear discrete-time systems (14) was published as a state prediction algorithm in the frame of a predictor–corrector state estimator in [26]. As visualized in Fig. 2, this procedure is based on propagating a thick ellipsoid $\left(\!\left(\check{\mathcal{E}}\right)\!\right)_k$ centered at the origin of the state space in parallel to an offset term (arising from non-zero ellipsoid midpoints $\boldsymbol{\mu}_k$) in the form

$$\mathbf{x}_{k+1} = \mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right) \cdot \check{\mathbf{x}}_k + \tilde{\mathbf{A}}_k \cdot \boldsymbol{\mu}_k + \left(\mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right) - \tilde{\mathbf{A}}_k \right) \cdot \boldsymbol{\mu}_k \, , \tag{15}$$

where

$$\left(\!\left(\mathcal{E}\right)\!\right)_k = \left(\!\left(\mathcal{E}\right)\!\right)_k \left(\boldsymbol{\mu}_k, \boldsymbol{\Gamma}_k, \left[\underline{\rho}_k \; ; \; \overline{\rho}_k\right]\right) \tag{16}$$

denotes the uncertainty on the non-origin centered states $\mathbf{x}_k$,

$$\left(\!\left(\check{\mathcal{E}}\right)\!\right)_k = \left(\!\left(\check{\mathcal{E}}\right)\!\right)_k \left(\mathbf{0}, \boldsymbol{\Gamma}_k, \left[\underline{\rho}_k \; ; \; \overline{\rho}_k\right]\right) \tag{17}$$

the uncertainty of $\check{\mathbf{x}}_k$ after shifting the ellipsoid to the origin, and

$$\tilde{\mathbf{A}}_k = \mathbf{A}\left(\boldsymbol{\mu}_k, \operatorname{mid}\left([\mathbf{p}_k]\right)\right) \tag{18}$$

the midpoint approximation of the quasi-linear system matrix with

$$\mathbf{p}_k \in [\mathbf{p}_k] = \left[\underline{\mathbf{p}}_k \; ; \; \overline{\mathbf{p}}_k\right], \quad \text{where} \quad \operatorname{mid}\left([\mathbf{p}_k]\right) = \frac{1}{2} \cdot \left(\underline{\mathbf{p}}_k + \overline{\mathbf{p}}_k\right) \; . \tag{19}$$

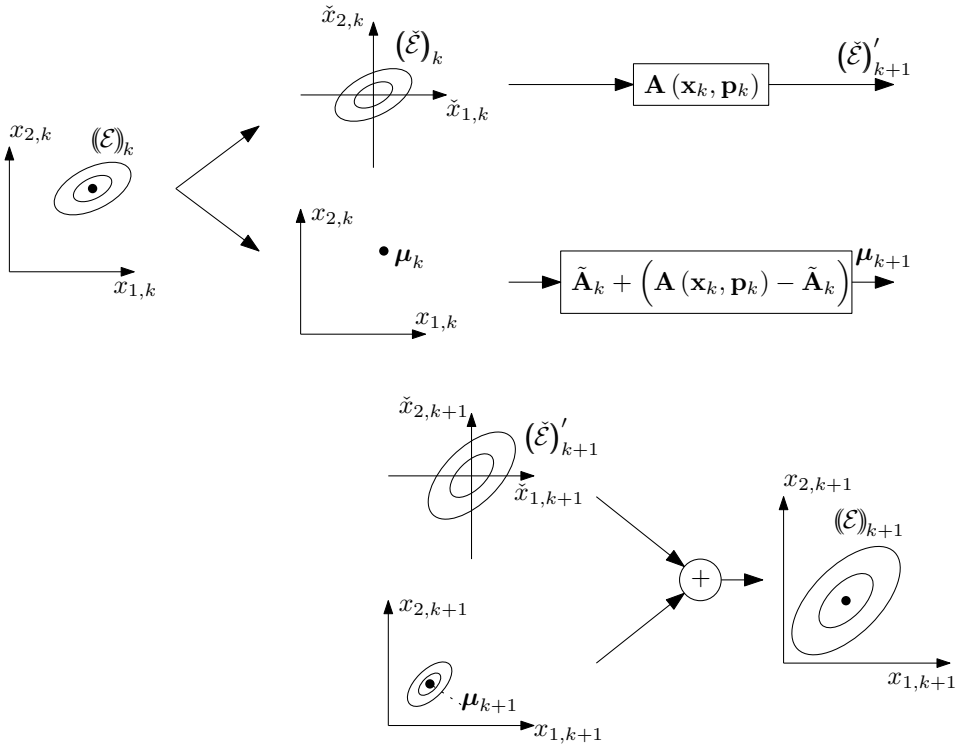


Figure 2: Separation of the state equations according to (15)–(19) into the mapping of an origin-centered ellipsoid and the verified treatment of non-zero offset terms.

1. Apply the mapping

$$\check{\mathbf{x}}_{k+1} = \mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right) \cdot \check{\mathbf{x}}_k \ , \tag{20}$$

with $\mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right)$ evaluated for all $\mathbf{x}_k \in \mathscr{E}_k^{\mathrm{O}}$ and $\mathbf{p}_k \in [\mathbf{p}_k]$, to the inner bound of $\left(\!\!\left(\check{\mathscr{E}}\right)\!\!\right)_k$ in (17). The shape matrix of the inner hull of the image set is given by

$$\check{\mathbf{Q}}_{k+1}^{\mathrm{I}} = \alpha_{\mathrm{I},k+1}^2 \cdot \underline{\rho}_k^2 \cdot \boldsymbol{\Gamma}_{k+1} \cdot \boldsymbol{\Gamma}_{k+1}^T \ , \tag{21}$$

where $\alpha_{\mathrm{I},k+1} \geq 0$ is the maximum value for which

$$\begin{aligned} \boldsymbol{\mathcal{M}}_{k+1} &:= \boldsymbol{\Lambda} \begin{bmatrix} \alpha_{\mathrm{I},k+1}^{-2} \cdot \boldsymbol{\mathscr{R}}_k^{-1} & \left(\tilde{\mathbf{A}}_k^{-1} \cdot \mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right)\right)^{-T} \\ \left(\tilde{\mathbf{A}}_k^{-1} \cdot \mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right)\right)^{-1} & \mathbf{Q}_k \end{bmatrix} \boldsymbol{\Lambda} \succeq 0 \ , \\ \mathbf{Q}_k &= \underline{\rho}_k^2 \cdot \boldsymbol{\Gamma}_k \cdot \boldsymbol{\Gamma}_k^T \end{aligned} \tag{22}$$

is satisfied in terms of positive semi-definiteness with the typical choice

$$\boldsymbol{\mathscr{R}}_k := \underline{\rho}_k^2 \cdot \boldsymbol{\Gamma}_k \cdot \boldsymbol{\Gamma}_k^T \ , \tag{23}$$

cf. [26]. An alternative choice for this matrix would be

$$\boldsymbol{\mathscr{R}}_k := \underline{\rho}_k^2 \cdot \tilde{\mathbf{A}}_k^{-T} \cdot \boldsymbol{\Gamma}_k \cdot \boldsymbol{\Gamma}_k^T \cdot \tilde{\mathbf{A}}_k^{-1} \ , \tag{24}$$

leading to a predicted ellipsoid that has an outer surface parallel to the one describing the state enclosure at the previous time step. As shown in [26], the option (24) is beneficial if the ratios of the lengths of the principal axes of the predicted ellipsoid differ significantly from the principal axes ratio of the original one. Note that the shape matrix definition (24) also simplifies the test for contractance in the following section.

As a generalization of the procedure derived in [26], the symmetric preconditioning matrix $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T \succ 0$ is introduced in (22). It helps to optimize the ellipsoidal enclosures, especially for the propagation of small state domains, i.e., if the norms of $\left(\tilde{\mathbf{A}}_k^{-1} \cdot \mathbf{A}\left(\mathbf{x}_k, \mathbf{p}_k\right)\right)^{-T}$ and $\mathbf{Q}_k$ are significantly different. Then, the non-rescaled equation with $\boldsymbol{\Lambda} = \mathbf{I}$ may be too conservative and yield unnecessarily empty inner bounds[1]. For rescaling purposes, a block diagonal matrix $\boldsymbol{\Lambda} = \mathrm{blkdiag}\left(\beta\mathbf{I}, \beta^{-1}\mathbf{I}\right)$ with $\mathbf{I} \in \mathbb{R}^{n \times n}$ and the square root $\beta = \sqrt{\min\{\lambda_i\left(\mathbf{Q}_k\right)\}}$ of the smallest eigenvalue of $\mathbf{Q}_k$ is used in this paper.

2. Apply (20) to the outer bound of $\left(\!\!\left(\check{\mathscr{E}}\right)\!\!\right)_k$ in (17). The shape matrix of the outer hull of the image set is given by

$$\check{\mathbf{Q}}_{k+1}^{\mathrm{O}} = \alpha_{\mathrm{O},k+1}^2 \cdot \overline{\rho}_k^2 \cdot \boldsymbol{\Gamma}_{k+1} \cdot \boldsymbol{\Gamma}_{k+1}^T \ , \tag{25}$$

[1] Omitting this rescaling in the following computation of outer bounds may also turn the results unnecessarily wide and less useful when applied in the frame of proving stability.

where $\alpha_{\mathrm{O},k+1} \geq 0$ is the smallest value for which

$$\begin{aligned} \mathscr{M}_{k+1} &:= \mathbf{\Lambda} \begin{bmatrix} -\mathbf{Q}_k^{-1} & \mathbf{A}^T(\mathbf{x}_k, \mathbf{p}_k) \cdot \tilde{\mathbf{A}}_k^{-T} \\ \tilde{\mathbf{A}}_k^{-1} \cdot \mathbf{A}(\mathbf{x}_k, \mathbf{p}_k) & -\alpha_{\mathrm{O},k+1}^2 \mathscr{R}_k \end{bmatrix} \mathbf{\Lambda} \preceq 0 \ , \\ \mathbf{Q}_k &= \overline{\rho}_k^2 \cdot \mathbf{\Gamma}_k \cdot \mathbf{\Gamma}_k^T \end{aligned} \tag{26}$$

is satisfied for all $\mathbf{x}_k \in \mathscr{E}_k^{\mathrm{O}}$ and $\mathbf{p}_k \in [\mathbf{p}_k]$ with $\mathscr{R}_k := \overline{\rho}_k^2 \cdot \mathbf{\Gamma}_k \cdot \mathbf{\Gamma}_k^T$.

3. Compute interval bounds for the term

$$\mathbf{b}_k = \left(\mathbf{A}(\mathbf{x}_k, \mathbf{p}_k) - \tilde{\mathbf{A}}_k \right) \cdot \boldsymbol{\mu}_k \in [\mathbf{b}_k] \tag{27}$$

which accounts for a non-zero ellipsoid midpoint with $\mathbf{x}_k$, $\tilde{\mathbf{A}}_k$, and $\mathbf{p}_k$ defined according to (16), (18), and (19). Deflate the inner ellipsoid bound from (21) according to

$$\mathbf{Q}_{k+1}^{\mathrm{I}} = \left(1 - \rho_{\mathrm{I},k+1}\right)^2 \cdot \check{\mathbf{Q}}_{k+1}^{\mathrm{I}} \ , \quad \rho_{\mathrm{I},k+1} = \sup \left\{ \left\| \alpha_{\mathrm{I},k+1}^{-1} \cdot \underline{\rho}_k^{-1} \cdot \mathbf{\Gamma}_k^{-1} \cdot [\mathbf{b}_k] \right\| \right\} \tag{28}$$

and inflate the outer bound in (25) with

$$\mathbf{Q}_{k+1}^{\mathrm{O}} = \left(1 + \rho_{\mathrm{O},k+1}\right)^2 \cdot \check{\mathbf{Q}}_{k+1}^{\mathrm{O}} \ , \quad \rho_{\mathrm{O},k+1} = \sup \left\{ \left\| \alpha_{\mathrm{O},k+1}^{-1} \cdot \overline{\rho}_k^{-1} \cdot \mathbf{\Gamma}_k^{-1} \cdot [\mathbf{b}_k] \right\| \right\} \ . \tag{29}$$

For $\rho_{\mathrm{I},k+1} \geq 1$, or if $\mathbf{A}(\mathbf{x}_k, \mathbf{p}_k)$ contains points at which it is not invertible, the inner bound becomes the empty set.

4. Compute the updated ellipsoid midpoint as

$$\boldsymbol{\mu}_{k+1} = \tilde{\mathbf{A}}_k \cdot \boldsymbol{\mu}_k \ . \tag{30}$$

5. The thick ellipsoid at the time instant $k+1$ then becomes

$$\mathbf{x}_{k+1} \in \left(\!\left(\mathscr{E}\right)\!\right)_{k+1} \left(\boldsymbol{\mu}_{k+1}, \mathbf{\Gamma}_{k+1}, \left[\underline{\rho}_{k+1} \ ; \ \overline{\rho}_{k+1} \right] \right) \ , \tag{31}$$

where

$$\begin{aligned} \underline{\rho}_{k+1} &= \underline{\rho}_k \cdot \alpha_{\mathrm{I},k+1} \cdot \left(1 - \rho_{\mathrm{I},k+1}\right) \ , \\ \overline{\rho}_{k+1} &= \overline{\rho}_k \cdot \alpha_{\mathrm{O},k+1} \cdot \left(1 + \rho_{\mathrm{O},k+1}\right) \ , \quad \text{and} \\ \mathbf{\Gamma}_{k+1} &= \tilde{\mathbf{A}}_k \cdot \mathbf{\Gamma}_k \ . \end{aligned} \tag{32}$$

Remark 3. For eigenvalue tests according to [36] as well as a Gershgorin circle criterion [47] that both allow for checking the definiteness properties in (22) and (26), the reader is referred to [26].

3 Ellipsoidal Stability Contractor

3.1 Specification of the Initial State Domain

For what follows, we assume that a linearization of the system model (4) at the equilibrium point $\mathbf{x}_0 = \mathbf{0}$ is given by the Jacobian $\mathbf{J}_0 = \mathbf{J}(\mathbf{0})$, where

$$\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}) \quad . \tag{33}$$

For the quasi-linear model (14) with its equilibrium at the origin of the state space, $\mathbf{J}_0$ is chosen as

$$\mathbf{J}_0 = \mathbf{A}\left(\mathbf{0}, \mathrm{mid}\left([\mathbf{p}]\right)\right) \quad , \tag{34}$$

where $[\mathbf{p}]$ denotes an interval box containing all (temporally constant) uncertain system parameters.

3.1.1 Point-Valued Selection Approach

To find ellipsoidal domains as enclosures of the initial conditions, for which the likelihood of convergence to the equilibrium state is as large as possible, we do not purely assume axis-aligned initial state domains but rather exploit the local dynamics properties of the (linearized) system model.

In the simplest approach, a reasonable shape matrix for the initial ellipsoidal state domain can be determined by solving the discrete-time Lyapunov equation

$$\mathbf{J}_0^T \mathbf{P} \mathbf{J}_0 - \mathbf{P} = -\mathbf{I} \ . \tag{35}$$

Here, the actual choice of the matrix on the right-hand side represents a degree of freedom with the prerequisite to be negative definite (in (35), the negative identity matrix $-\mathbf{I}$ is used). To avoid specifying this matrix explicitly, the equality (35) can be cast equivalently into the LMI

$$\mathbf{J}_0^T \mathbf{P} \mathbf{J}_0 - \mathbf{P} \prec 0 \tag{36}$$

for which a solution $\mathbf{P} = \mathbf{P}^T \succ 0$ needs to be found. In analogy to a positive definite solution $\mathbf{P}$ of the Lyapunov equation (35), the existence of a solution to the LMI (36) corresponds to the local asymptotic stability of the linearized system model at the origin of the state space. The ellipsoid shape matrix $\mathbf{Q} = \mathbf{\Gamma} \cdot \mathbf{\Gamma}^T$ is then obtained according to the matrix inverse

$$\mathbf{Q} = \mathbf{P}^{-1} \ . \tag{37}$$

3.1.2 Robust Domain Specification

The drawback of the ellipsoid parameterization according to (35) and (36) is the fact that both approaches only take into account a point-valued system model in terms of a linearization at the equilibrium state. This restriction can be removed if

a polytopic uncertainty model is derived such that $\mathbf{J}(\mathbf{x})$ and $\mathbf{A}(\mathbf{x}, \mathbf{p})$ with $\mathbf{p} \in [\mathbf{p}]$, respectively, are bounded by the convex polytopic domain

$$\mathbf{J}(\mathbf{x}) \in \left\{ \mathbf{J} \middle| \mathbf{J}(\xi) = \mathbf{J}_0' + \sum_{v=1}^{n_v} \xi_v \cdot \Delta\mathbf{J}_v \; ; \quad \sum_{v=1}^{n_v} \xi_v = 1 \; ; \quad \xi_v \geq 0 \right\} \; . \tag{38}$$

Here, $\mathbf{x}$ needs to be replaced with a set-valued representation that encloses an application-motivated domain of interest for which stability shall be investigated. The domain (38) is spanned by a collection of at most $n_v = 2^{n^2}$ vertices, where the worst-case deviations of all possible realizations of the Jacobian $\mathbf{J}(\mathbf{x})$ from $\mathbf{J}_0'$ are described by the matrices $\Delta\mathbf{J}_v$. In (38), the matrix $\mathbf{J}_0'$ is a point-valued matrix included in the set-based evaluation of $\mathbf{J}(\mathbf{x})$, which is enclosed by a convex polytope that is spanned with the help of the individual increment matrices $\Delta\mathbf{J}_v$.

Using this formulation, (36) can be replaced in a conservative manner by the collection of LMIs

$$\left(\mathbf{J}_0' + \Delta\mathbf{J}_v\right)^T \cdot \mathbf{P} \cdot \left(\mathbf{J}_0' + \Delta\mathbf{J}_v\right) - \mathbf{P} \prec 0 \tag{39}$$

for which a joint solution $\mathbf{P} = \mathbf{P}^T \succ 0$ in terms of a vertex-independent quadratic Lyapunov function parameterization needs to be found. The existence of such a matrix $\mathbf{P}$ proves that each vertex realization, and hence all convex combinations of vertices according to (38), correspond to asymptotically stable realizations.

It should be pointed out that with the help of a quadratic Lyapunov function candidate $\mathbf{x}^T \cdot \mathbf{P} \cdot \mathbf{x}$ — that is parameterized according to (39) — only a proof of asymptotic stability for states satisfying the inequality

$$\mathbf{f}^T(\mathbf{x}_0) \cdot \mathbf{P} \cdot \mathbf{f}(\mathbf{x}_0) - \mathbf{x}_0^T \cdot \mathbf{P} \cdot \mathbf{x}_0 < 0 \tag{40}$$

in the interior of a contour line $\mathbf{x}_0^T \cdot \mathbf{P} \cdot \mathbf{x}_0 = c$, $c > 0$ that is fully included in the box $\mathbf{x}_0 \in [\mathbf{x}_0]$ is obtained. Note that this interval box needs to be employed for generating the polytopic uncertainty representation (38). Hence, the direct application of a stability contractor to the system models (4) and (14), making use of an ellipsoid with a shape matrix $\mathbf{Q}$ computed by (37) simplifies the evaluation of (40) and provides reasonably large provable stability domains as long as the contractor itself can be evaluated with a small amount of overestimation.

The major drawback of the polytopic uncertainty representation (38) is the typically large number of vertices that results from treating each matrix entry of the Jacobian (or of the quasi-linear system matrix, respectively) as independent.

Remark 4. The number n_v of the vertices to be considered in the polytopic uncertainty representation (38) can often be reduced by identifying physically motivated linear dependencies between individual entries of the matrix $\mathbf{J}(\mathbf{x})$ and by expressing them in terms of common interval parameters. For an example, where this has been done successfully, the reader is referred to [11].

When determining a candidate for the shape matrix of the initial state domain, in which stability is investigated, a further reduction of the complexity can be obtained by the introduction of a norm-bounded uncertainty model

$$\mathbf{J}\left(\mathbf{x}\right) \in \mathbf{J}_0 + \Delta\mathbf{J} \ , \quad \text{where} \quad \Delta\mathbf{J} = \mathbf{H} \cdot \mathbf{F} \cdot \mathbf{E} \tag{41}$$

holds with $\mathbf{F}$ being an unknown, norm-bounded matrix according to $\|\mathbf{F}\| < 1$. The example in Sec. 3.4 demonstrates how the matrices $\mathbf{E}$ and $\mathbf{H}$ included in (41) can be chosen to represent the variability of the matrix $\mathbf{J}\left(\mathbf{x}\right)$ over the investigated domain. There, the simplest choice is shown by setting one of the matrices to the identity matrix, and to define the second as the worst-case interval radii of each element of $\mathbf{J}\left(\mathbf{x}\right)$ if $\mathbf{J}_0$ is set to the element-wise defined interval midpoint. Using this norm-bounded model, a single LMI needs to be solved instead of finding a common solution to the previous list of n_v matrix inequalities.

Stability of the norm-bounded uncertainty model is verified according (39). Rewriting this inequality by applying the Schur complement formula leads to

$$\begin{bmatrix} -\mathbf{P} & \mathbf{J}_0^T \\ \mathbf{J}_0 & -\mathbf{P}^{-1} \end{bmatrix} + \begin{bmatrix} \mathbf{E}^T \\ \mathbf{0} \end{bmatrix} \mathbf{F}^T \begin{bmatrix} \mathbf{0} & \mathbf{H}^T \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} \mathbf{F} \begin{bmatrix} \mathbf{E} & \mathbf{0} \end{bmatrix} \prec 0 \ , \quad \mathbf{P} \succ 0 \ . \tag{42}$$

Then, the application of the elimination lemma [40] allows for eliminating the unknown matrix $\mathbf{F}$. It turns the nonlinear matrix inequality[2] (42) into

$$\begin{bmatrix} -\mathbf{P} & \mathbf{J}_0^T \\ \mathbf{J}_0 & -\mathbf{P}^{-1} \end{bmatrix} + \epsilon^{-1} \begin{bmatrix} \mathbf{E}^T \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{0} \end{bmatrix} + \epsilon \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{H}^T \end{bmatrix} \prec 0 \ , \quad \mathbf{P} \succ 0 \ , \tag{43}$$

where $\epsilon > 0$ is a free parameter. After combining the second and third terms of the inequality (43), it becomes equivalent to

$$\begin{bmatrix} -\mathbf{P} & \mathbf{J}_0^T \\ \mathbf{J}_0 & -\mathbf{P}^{-1} \end{bmatrix} + \begin{bmatrix} \mathbf{E}^T \\ \epsilon\mathbf{H} \end{bmatrix} \epsilon^{-1}\mathbf{I} \begin{bmatrix} \mathbf{E} & \epsilon\mathbf{H}^T \end{bmatrix} \prec 0 \ , \quad \mathbf{P} \succ 0 \ , \tag{44}$$

which can be transformed by applying the Schur complement into

$$\begin{bmatrix} -\mathbf{P} & \mathbf{J}_0^T & \mathbf{E}^T \\ \mathbf{J}_0 & -\mathbf{P}^{-1} & \epsilon\mathbf{H} \\ \mathbf{E} & \epsilon\mathbf{H}^T & \epsilon\mathbf{I} \end{bmatrix} \prec 0 \ , \quad \mathbf{P} \succ 0 \ . \tag{45}$$

After multiplication of the matrix inequality (45) from the left and right with the block diagonal matrix

$$\text{blkdiag}\left(\mathbf{P}^{-1}, \mathbf{I}, \mathbf{I}\right) =: \text{blkdiag}\left(\mathbf{Q}, \mathbf{I}, \mathbf{I}\right) \ , \tag{46}$$

[2] due to the inverse of the decision variable matrix $\mathbf{P}$

the LMI formulation

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{Q}\mathbf{J}_0^T & \mathbf{Q}\mathbf{E}^T \\ \mathbf{J}_0\mathbf{Q} & -\mathbf{Q} & \epsilon\mathbf{H} \\ \mathbf{E}\mathbf{Q} & \epsilon\mathbf{H}^T & \epsilon\mathbf{I} \end{bmatrix} \prec 0 \ , \quad \mathbf{Q} \succ 0 \tag{47}$$

is obtained. It verifies asymptotic stability of the norm-bounded uncertainty model (41) if $\mathbf{Q} = \mathbf{Q}^T \succ 0$ and, therefore, $\mathbf{P} = \mathbf{P}^T \succ 0$ exists, where the actual value of $\epsilon > 0$ can be determined automatically by the LMI solver.

3.2 Verification of the Property of Contractance

For the verification of the property of contractance in the case of a thick ellipsoid stability check, we assume that the prior state domain (given as a crisp ellipsoid $\mathscr{E}_0 = (\!(\mathscr{E})\!)_0 \left(\mathbf{0}, \mathbf{\Gamma}_0, [\rho_0 \ ; \ \rho_0]\right)$ with identical outer and inner bounds and a shape matrix parameterized according to the options listed in the previous subsection) is centered at the equilibrium state of the system (the origin, without loss of generality) and that it is mapped onto a thick ellipsoid $(\!(\mathscr{E})\!)_1 \left(\mathbf{0}, \mathbf{\Gamma}_1, \left[\underline{\rho}_1 \ ; \ \overline{\rho}_1\right]\right)$ that is again centered at the equilibrium.

Then, it is guaranteed by a one time step evaluation of the system model, that $\mathscr{E}_0$ belongs to the region of attraction of the equilibrium if the predicted outer bound $\mathscr{E}_1^{\mathrm{O}}$ is a guaranteed subset of $\mathscr{E}_0$ according to

$$\mathscr{E}_1^{\mathrm{O}} \subset \mathscr{E}_0 \ . \tag{48}$$

The property (48) can be checked by verifying whether *all* eigenvalues λ_i of the shape matrix difference satisfy the inequality[3]

$$\lambda_i = \lambda_i \left(\left(\overline{\rho}_1^2 \cdot \mathbf{\Gamma}_1 \cdot \mathbf{\Gamma}_1^T \right)^{-1} - \left(\rho_0^2 \cdot \mathbf{\Gamma}_0 \cdot \mathbf{\Gamma}_0^T \right)^{-1} \right) > 0 \ . \tag{49}$$

This inequality is a direct consequence of the proof of Theorem 3 in [26].

In contrast, if

$$\lambda_i = \lambda_i \left(\left(\underline{\rho}_1^2 \cdot \mathbf{\Gamma}_1 \cdot \mathbf{\Gamma}_1^T \right)^{-1} - \left(\rho_0^2 \cdot \mathbf{\Gamma}_0 \cdot \mathbf{\Gamma}_0^T \right)^{-1} \right) < 0 \tag{50}$$

holds for *all* of the eigenvalues according to the proof of Theorem 1 in [26], it is guaranteed that the domain $\mathscr{E}_0$ is an unstable neighborhood of the equilibrium $\mathbf{x}_0$ according to Chetaev's theorem, see [19, Theorem 3.12]. Geometrically, this

[3] A rigorous proof of the inequalities (49) and (50) is possible with the help of the routine `verifyeig` included in IntLab [39]. Alternatively, the matrices can be diagonalized as far as possible using verified numerics with a subsequent eigenvalue test following Remark 3. In many practical cases, however, it often suffices to check in classical floating point arithmetic whether the eigenvalues with smallest magnitude have a sufficiently large distance to the value zero.

corresponds to the fact that the predicted inner thick ellipsoid bound fully encloses the sufficiently small original domain according to

$$\mathscr{E}_1^{\mathrm{I}} \supset \mathscr{E}_0 \quad . \tag{51}$$

Note, this case only arises for an intuitive choice of $\mathscr{E}_0$ (cf. Sec. 3.5) because it contradicts the robust LMI constraints listed above. Moreover, it should be noted that the difference $\overline{\rho}_1 - \underline{\rho}_1$ directly serves as a quantification of the possible overestimation of the predicted ellipsoid hulls according to [29].

The check of the eigenvalue inequalities (49) and (50) is necessary when applying either the general-purpose ellipsoidal enclosures according to Sec. 2.3 (Theorem 1) or when using the quasi-linear formulation (Sec. 2.4) with (23) as the parameterization for the predicted shape matrix. If the simplification (24) is used in the case of Sec. 2.4, the inequality (49) turns into $\overline{\rho}_1 < \rho_0$ and (50) turns into $\underline{\rho}_1 > \rho_0$.

To maximize the domains for which stability can be proven by this contractor, the examples in the following two subsections try to find the largest positive value ρ by means of a bisection algorithm so that $\min(\lambda_i) > 0$ holds in (49), where the threshold $\epsilon^* = 10^{-6}$ is used as the tolerance between two subsequent admissible solutions for the parameter ρ.

3.3 Proof of Stability: An Illustrating Example

As a first illustrating example, consider an explicit Euler discretization of the second-order system model $\dot{x}_1 = -x_1$, $\dot{x}_2 = -x_2 + x_1^2 x_2$ that was used in [15, 44] as a numerical benchmark scenario for the analysis of continuous-time ordinary differential equations. The discrete-time state equations can be specified as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + T \cdot \begin{bmatrix} -x_{1,k} \\ -x_{2,k} + x_{1,k}^2 x_{2,k} \end{bmatrix} \quad \text{leading to} \quad \mathbf{J}_0 = \begin{bmatrix} 1-T & 0 \\ 0 & 1-T \end{bmatrix} \quad , \tag{52}$$

which can be re-written (with the unique equilibrium $\mathbf{x}_0 = \mathbf{0}$) into the quasi-linear form

$$\mathbf{x}_{k+1} = \mathbf{A}(\mathbf{x}_k) \cdot \mathbf{x}_k = \begin{bmatrix} 1-T & 0 \\ T\alpha x_{1,k} x_{2,k} & (1-T) + T(1-\alpha) \cdot x_{1,k}^2 \end{bmatrix} \cdot \mathbf{x}_k \quad \text{with} \quad \alpha \in \mathbb{R} \quad . \tag{53}$$

For this special example, $\mathbf{J}_0 = \mathbf{A}(\mathbf{0})$ holds. In general, the quasi-linear reformulation (53) is not unique. Therefore, the parameter α can be used as an optimization variable (in addition to the parameter ρ) of the initial ellipsoidal state domain in order to maximize the provable domain of attraction of the equilibrium.

Fig. 3 gives an overview of the provable stability domains by means of a symbolic evaluation of the discrete-time Lyapunov function increment (40) for a one time step evaluation of the system model. In Fig. 3, the result **A** denotes the maximum provable domain with the given Lyapunov function candidate; moreover, the general nonlinear ellipsoidal enclosure technique (result **B**), the quasi-linear ellipsoid

implementation (result **C**) as well as an interval-based contractor implementation (result **D**) are compared. Due to the fact that the system model has a dominant linear behavior in the close vicinity to the equilibrium, the quasi-linear evaluation outperforms the general nonlinear technique. Moreover, it can be seen that the actual choice of the parameter α has a strong influence on the volume of the provable stability domain, where the maximum-volume ellipsoid is close to the volume of the largest provable box volume in the case of Figs. 3a and 3b and even larger for the heuristic choice of Figs. 3c and 3d. Note that the visible spikes in the volume dependency can be removed by slightly adapting the scaling matrix $\mathbf{\Lambda}$ in (26).

Due to the fact that all domains shown in the left column of Fig. 3 are guaranteed to contain asymptotically stable system realizations, their set-valued union can be formed to describe the domain in the state space for which the system exhibits asymptotically stable dynamics. The fact that the quasi-linear contractor outperforms the general nonlinear ellipsoidal enclosures gives rise to the following aspect of future work: Find a unified implementation for both approaches in which the quasi-linear system matrix and/or a suitable interval extension in slope arithmetic [6, 38] are employed to enhance the tightness of solutions. Note that the interval contractor was evaluated in an overestimation-free manner for this example after a symbolic reformulation of the state equations. This fact emphasizes the advantageous property of the ellipsoidal approach in Figs. 3a and 3c to prove stability of initial conditions that could not be detected by the interval counterpart for the same choice of aspect ratio (resulting from the precomputed ellipsoid shape matrix $\mathbf{Q}$).

In Figs. 3e and 3f, however, it can be seen that the ellipsoid enclosures are much smaller than the interval contractor's volume. This is caused by the fact that the included matrices $\mathbf{J}(\mathbf{x})$ need to be evaluated on a box that encloses the ellipsoid domain from the outside which leads to a kind of wrapping effect. For this specific setting of the shape matrix $\mathbf{Q}$, parts of those domains are close to the stability boundary so that the ellipsoid approach performs worse than the interval-based counterpart. In such cases, the approaches included in [13,14] for the computation of outer state enclosures could be helpful to enhance the procedures of [3]. In general, however, the ellipsoidal approach will be more efficient if the domains under investigation are chosen on the basis of Lyapunov function candidates.

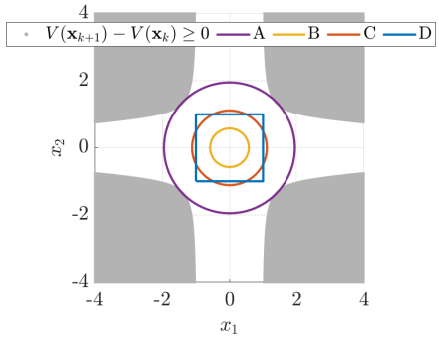
3.4 Stability Proof of an EKF-Based Localization Algorithm

As a second application scenario, we re-consider the stability proof of an EKF-based localization algorithm, for which an interval-based stability contractor was investigated in [3].

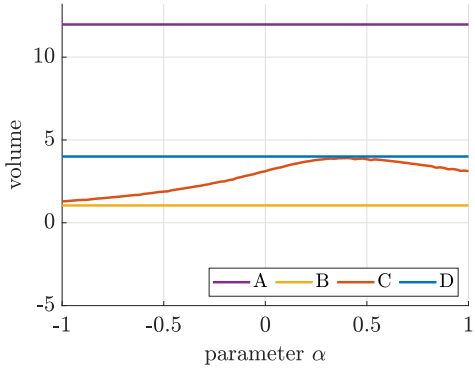
For this scenario, the output equation is given by

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} (x_{1,k} - a_1)^2 + (x_{2,k} - a_2)^2 \\ (x_{1,k} - b_1)^2 + (x_{2,k} - b_2)^2 \end{bmatrix} \quad , \tag{54}$$

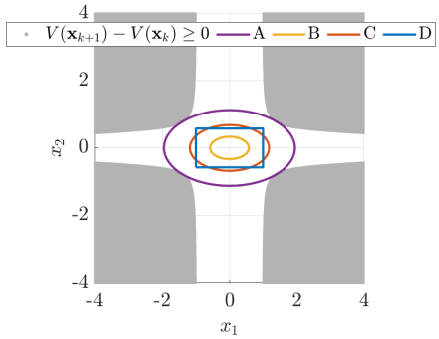
where (a_1, a_2) and (b_1, b_2) denote the known positions of two beacons; the vector $\mathbf{y}_k$ denotes the squared distances to the object $\mathbf{x}_k$ to be localized. This measurement



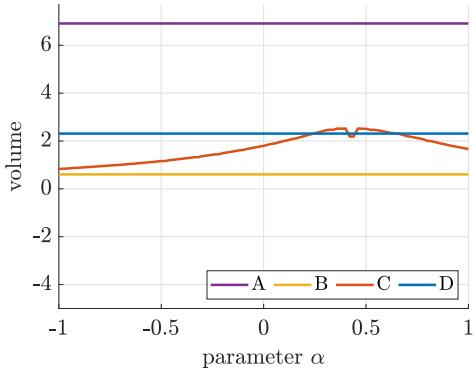
(a) Selection of $\mathbf{P} = \mathbf{Q}^{-1}$ acc. to (35), (36).



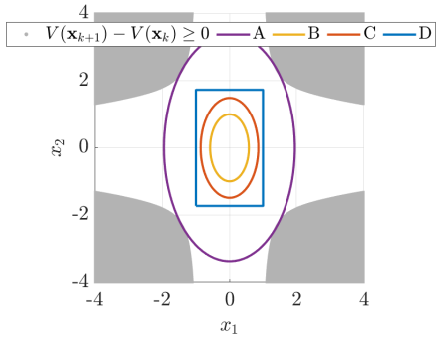
(b) Selection of $\mathbf{P} = \mathbf{Q}^{-1}$ acc. to (35), (36).



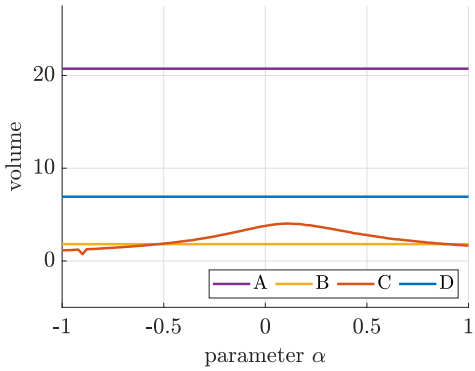
(c) Selection of $\mathbf{P} = \mathbf{Q}^{-1} = \text{diag}\left(\begin{bmatrix}1 & 3\end{bmatrix}\right)$.



(d) Selection of $\mathbf{P} = \mathbf{Q}^{-1} = \text{diag}\left(\begin{bmatrix}1 & 3\end{bmatrix}\right)$.



(e) Selection of $\mathbf{P} = \mathbf{Q}^{-1} = \text{diag}\left(\begin{bmatrix}3 & 1\end{bmatrix}\right)$.



(f) Selection of $\mathbf{P} = \mathbf{Q}^{-1} = \text{diag}\left(\begin{bmatrix}3 & 1\end{bmatrix}\right)$.

Figure 3: Provable stability domains for the example (52), (53) in the left column (using the parameter α with the largest ellipsoid volume in the case **C**) and dependence of the volume of the provable stability domain on the parameter $\alpha \in [-1\ ;\ 1]$ (right column).

(subscript m) is assumed to be corrupted by additive, zero-mean Gaussian noise $\mathbf{v}_k$ with the covariance $\mathbf{C}_v$, so that $\mathbf{y}_{\mathrm{m},k} = \mathbf{y}_k + \mathbf{v}_k$ holds. Moreover, we assume that the object to be localized is described by a discrete-time integrator disturbance model with additive, zero-mean Gaussian process noise $\mathbf{w}_k$ according to

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \mathbf{w}_k \ , \tag{55}$$

where the disturbance covariance matrix related to $\mathbf{w}_k$ is denoted by $\mathbf{C}_w$.

Then, an EKF algorithm can be specified with the help of the augmented state vector

$$\mathbf{x}_k = \begin{bmatrix} x_{1,k} & x_{2,k} & c_{11,k} & c_{12,k} & c_{22,k} \end{bmatrix}^T \ , \tag{56}$$

which consists of the estimated position $(x_{1,k}, x_{2,k})$ and the entries of the covariance matrix $\mathbf{C}_k$ after the innovation stage at the time instant k. The position and covariance matrix can be extracted with the help of selection matrices

$$\mathbf{S}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{S}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \ , \tag{57}$$

where

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = \mathbf{S}_1 \cdot \mathbf{x}_k \tag{58}$$

and

$$\begin{bmatrix} c_{11,k} \\ c_{12,k} \\ c_{22,k} \end{bmatrix} = \mathrm{vech}\left(\mathbf{C}_k\right) = \mathbf{S}_2 \cdot \mathbf{x}_k \ , \ \ \mathbf{C}_k = \begin{bmatrix} c_{11,k} & c_{12,k} \\ c_{12,k} & c_{22,k} \end{bmatrix} \iff \mathbf{C}_k = \mathrm{vech}^{-1}\left(\mathbf{S}_2 \cdot \mathbf{x}_k\right) \ . \tag{59}$$

Here, extracting the upper triangular part of the covariance matrix $\mathbf{C}_k$ is performed by the half-vectorization operator vech, where the corresponding inverse operation is denoted by vech^{-1}. With this notation, the state equations of the EKF can be specified so that $\mathbf{x}_{k+1}$ contains the estimated position and covariance matrix entries after performing the subsequent prediction and innovation step associated with the time instant $k+1$. Hence, these equations are given by

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k\right) = \begin{bmatrix} \mathbf{S}_1 \cdot \mathbf{x}_k + \mathbf{K}\left(\mathbf{x}_k\right) \cdot \left(\mathbf{y}_{\mathrm{m},k} - \mathbf{h}\left(\mathbf{x}_k\right)\right) \\ \mathrm{vech}\left(\left(\mathbf{I} - \mathbf{K}\left(\mathbf{x}_k\right) \cdot \mathbf{H}\left(\mathbf{x}_k\right)\right) \cdot \mathbf{C}_{k+1}^{\mathrm{p}}\left(\mathbf{x}_k\right)\right) \end{bmatrix} \tag{60}$$

with the predicted covariance matrix

$$\mathbf{C}_{k+1}^{\mathrm{p}}\left(\mathbf{x}_k\right) = \mathrm{vech}^{-1}\left(\mathbf{S}_2 \cdot \mathbf{x}_k\right) + \mathbf{C}_w \ , \tag{61}$$

the Kalman gain

$$\mathbf{K}\left(\mathbf{x}_k\right) = \mathbf{C}_{k+1}^{\mathrm{p}}\left(\mathbf{x}_k\right) \cdot \mathbf{H}^T\left(\mathbf{x}_k\right) \cdot \left(\mathbf{H}\left(\mathbf{x}_k\right) \cdot \mathbf{C}_{k+1}^{\mathrm{p}}\left(\mathbf{x}_k\right) \cdot \mathbf{H}^T\left(\mathbf{x}_k\right) + \mathbf{C}_v\right)^{-1} \ , \tag{62}$$

and the Jacobian

$$\mathbf{H}\left(\mathbf{x}_k\right) = \left[\frac{\partial \mathbf{h}}{\partial x_{1,k}}\left(\mathbf{x}_k\right) \quad \frac{\partial \mathbf{h}}{\partial x_{2,k}}\left(\mathbf{x}_k\right)\right] \tag{63}$$

of the output equation with respect to the current position estimate. For the following numerical results in Tabs. 1 and 2, we consider the beacon positions $a_1 = -5$, $a_2 = 5$, $b_1 = 5$, $b_2 = 5$, the measurement $\mathbf{y}_{\mathrm{m},k} = \begin{bmatrix}0 & 0\end{bmatrix}^T$, and the noise covariances $\mathbf{C}_w = \mathrm{diag}\left(\begin{bmatrix}0.01 & 0.01\end{bmatrix}\right)$ as well as $\mathbf{C}_v = \mathrm{diag}\left(\begin{bmatrix}1 & 1\end{bmatrix}\right)$. This leads to the equilibrium state $x_1^* = x_2^* = 0$, $c_{12}^* = 0$ and $c_{11}^* = c_{22}^* = 0.003660254037844$ around which the stability domains are centered.

Using a point-valued selection of the ellipsoid shape matrix according to Sec. 3.1.1 leads to the result in Tab. 1 which can be widened by the robustified, norm-bounded uncertainty representation according to Sec. 3.1.2, see Tab. 2. For that purpose, the norm-bounded uncertainty model in (41) is parameterized by choosing $\mathbf{E} = \mathbf{I}$. Then, the matrix $\mathbf{H}$ is specified so that the maximum interval radii of an interval extension of the Jacobian $\mathbf{J}\left(\mathbf{x}\right)$ on a representative domain are captured in an element-wise sense by the additive term $\Delta\mathbf{J}$. It should be pointed out that the provably stable interval box domains are significantly larger than those reported in [3]. This is caused (i) by choosing the ratios of the interval edge lengths identical to the ratios of the edge lengths of an axis-aligned box corresponding to the Lyapunov function and LMI-based shape matrix definitions, and (ii) by not only using a centered form evaluation but also intersecting it with a slope extension of the range implemented in IntLab [38,39]. This kind of evaluation can also be integrated into the ellipsoidal approach in future work.

3.5 Proof of Instability: An Illustrating Example

To demonstrate the applicability of the ellipsoidal approach to find unstable neighborhoods of equilibrium points by means of (51) and the inequality (50), consider the explicit Euler discretization with $T = 0.1$ of the benchmark example (3.23) in [19] for which $\beta = 1$ is chosen. In a quasi-linear form, this example has the state equations

$$\mathbf{x}_{k+1} = \left(\mathbf{I} + T \cdot \begin{bmatrix}\beta^2 - x_{1,k}^2 - x_{2,k}^2 & 1 \\ -1 & \beta^2 - x_{1,k}^2 - x_{2,k}^2\end{bmatrix}\right) \cdot \mathbf{x}_k \ . \tag{64}$$

Parameterizing the initial state domain $\mathscr{E}_0$ as a circle with radius 0.1 leads to circles as the inner ellipsoidal enclosures $\mathscr{E}_1^{\mathrm{I}}$ with the inward rounded radii 0.10953174 and 0.11002215 for the general-purpose and the quasi-linear evaluation approaches of Secs. 2.3 and 2.4, respectively. Due to $\mathscr{E}_1^{\mathrm{I}} \supset \mathscr{E}_0$, the domain $\mathscr{E}_0$ is a provably unstable neighborhood of the equilibrium $\mathbf{x}_0 = \mathbf{0}$, where the quasi-linear approach provides the less conservative solution.

Table 1: Comparison of different stability contractors for a shape matrix selection according to Sec. 3.1.1; for the ellipsoid case, a tight outer, axis-aligned hull is given.

	interval contractor	
	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$
x_1	−0.00427481791343	0.00427481791343
x_2	−0.00427481599897	0.00427481599897
c_{11}	−0.00076536378418	0.00808587185987
c_{12}	−0.00442561782520	0.00442561782520
c_{22}	−0.00076536379689	0.00808587187258

	ellipsoidal encl. (quasi-lin.)	
	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$
x_1	−0.00161026895377	0.00161026895377
x_2	−0.00161026823262	0.00161026823262
c_{11}	0.00199318069709	0.00532732737860
c_{12}	−0.00166707334195	0.00166707334195
c_{22}	0.00199318069230	0.00532732738339

	ellipsoidal encl. (general.)	
	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$
x_1	−0.00043890766400	0.00043890766400
x_2	−0.00043890746744	0.00043890746744
c_{11}	0.00320586332007	0.00411464475562
c_{12}	−0.00045439071810	0.00045439071810
c_{22}	0.00320586331877	0.00411464475692

Table 2: Comparison of different stability contractors for a robustified shape matrix selection according to Sec. 3.1.2; for the ellipsoid case, a tight outer, axis-aligned hull is given.

	interval contractor	
	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$
x_1	−0.00440136214825	0.00440136214825
x_2	−0.00440079183131	0.00440079183131
c_{11}	−0.00076807607385	0.00808858414954
c_{12}	−0.00441865927874	0.00441865927874
c_{22}	−0.00076814256179	0.00808865063748

	ellipsoidal encl. (quasi-lin.)	
	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$
x_1	−0.00168345510722	0.00168345510722
x_2	−0.00168323696953	0.00168323696953
c_{11}	0.00196648408848	0.00535402398721
c_{12}	−0.00169007100059	0.00169007100059
c_{22}	0.00199318069230	0.00535404941785

	ellipsoidal encl. (general.)	
	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$
x_1	−0.00044551035825	0.00044551035825
x_2	−0.00044545263019	0.00044545263019
c_{11}	0.00321201395487	0.00410849412082
c_{12}	−0.00044726119141	0.00044726119141
c_{22}	0.00321200722490	0.00410850085079

4 Outlook: Ellipsoid Definition of Positive Invariant Sets for Continuous-Time Processes

Due to the fact that the ellipsoidal contractor above is based on a forward in time evaluation of dynamic system models, it is also readily applicable to the continuous-time case if the solution approach published in [27] is employed. However, to reduce pessimism, it can be combined in future work with the following novel test for positive invariance.

Theorem 2 (Positive invariant, ellipsoidal domains for continuous-time systems)**.** *Consider the continuous-time system* $\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)\right)$, $\mathbf{x} \in \mathbb{R}^n$ *with the (locally) stable equilibrium* $\mathbf{x} = \mathbf{0}$. *Define the Lyapunov function candidate*

$$V\left(\mathbf{x}(t)\right) = \frac{1}{2}\mathbf{x}^T \boldsymbol{\mathscr{P}} \mathbf{x} \ , \quad \mathbf{x} := \mathbf{x}(t) \tag{65}$$

with $\boldsymbol{\mathscr{P}} \succ 0$ *and the small interval box* $[\mathbf{x}] \ni \mathbf{0}$. *Define the ellipsoid*

$$\mathscr{E}_{\boldsymbol{\mathscr{P}}}\left([\mathbf{x}]\right) = \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; -\mathbf{x}^T \cdot \boldsymbol{\mathscr{P}} \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{0}) \cdot \mathbf{x} \leq v^+ \right\} \tag{66}$$

with

$$v^+ = \sup\left([\mathbf{x}]^T \cdot \boldsymbol{\mathscr{P}} \cdot [\mathbf{e}] \right) \quad \textit{and} \quad [e_i] = \frac{1}{2}[\mathbf{x}]^T \cdot \frac{\partial^2 f_i}{\partial \mathbf{x}^2}\left([\mathbf{x}]\right) \cdot [\mathbf{x}] \ . \tag{67}$$

If $\mathscr{E}_{\boldsymbol{\mathscr{P}}}\left([\mathbf{x}]\right) \subset [\mathbf{x}]$, $\mathscr{E}_{\boldsymbol{\mathscr{P}}}\left([\mathbf{x}]\right)$ *is positive invariant.*

Proof. Take an interval box $\mathbf{x}$ with center at $\mathbf{x} = \mathbf{0}$ and express the i-th component of $\mathbf{f}$ as a second-order Taylor form near the equilibrium, i.e.,

$$\mathbf{x} \in [\mathbf{x}] \Longrightarrow f_i(\mathbf{x}) \in J_{i:} \cdot \mathbf{x} + \frac{1}{2}[\mathbf{x}]^T \cdot [\mathbf{H}_i] \cdot [\mathbf{x}] = J_{i:} \cdot \mathbf{x} + [e_i] \ , \tag{68}$$

where $[\mathbf{H}_i]$ is an interval extension of the Hessian of f_i and $J_{i:}$ is the i-th row of the Jacobian $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{0})$. Consequently,

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \mathbf{x}^T \cdot \boldsymbol{\mathscr{P}} \cdot \mathbf{f}(\mathbf{x}) \\ &= \sum_{i=1}^{n} x_i \cdot \mathscr{P}_{:i} \cdot f_i(\mathbf{x}) \\ &\in \sum_{i=1}^{n} x_i \cdot \mathscr{P}_{:i} \cdot \left(J_{i:} \cdot \mathbf{x} + [e_i] \right) \\ &= \mathbf{x}^T \cdot \boldsymbol{\mathscr{P}} \cdot \mathbf{J} \cdot \mathbf{x} + \mathbf{x}^T \cdot \boldsymbol{\mathscr{P}} \cdot [\mathbf{e}] \ . \end{aligned} \tag{69}$$

Setting $[v] = [\mathbf{x}]^T \cdot \boldsymbol{\mathscr{P}} \cdot [\mathbf{e}]$, we have $\dot{V}(\mathbf{x}) < \mathbf{x}^T \cdot \boldsymbol{\mathscr{P}} \cdot \mathbf{J} \cdot \mathbf{x} + \sup\left([v]\right)$. Taking $\mathbf{x}$ such that $\mathbf{x}^T \cdot \boldsymbol{\mathscr{P}} \cdot \mathbf{J} \cdot \mathbf{x} + v^+ = 0$, where $v^+ \sup\left([v]\right)$, yields $\dot{V}(\mathbf{x}) < 0$ for $\mathbf{x} \in \mathscr{E}_{\boldsymbol{\mathscr{P}}}\left([\mathbf{x}]\right)$ according to (66), (67) which completes the proof. □

5 Conclusions

In this paper, an ellipsoidal implementation of a stability contractor was presented for discrete-time systems. Due to the possibility for finding initial parameterizations of the shape matrix by means of Lyapunov equations or LMIs, it has the advantage in comparison to a straightforward interval-based implementation that the considered domains are not necessarily axis-parallel and that the form of the domains investigated is close to (locally valid) Lyapunov function candidates. In such a way, it becomes possible to often find larger domains of attraction than for the previously investigated interval-based counterpart. In addition, it was shown that the use of a specialized implementation for quasi-linear system models may outperform the application of a general ellipsoidal enclosure technique. This is especially true if free parameters in the quasi-linear system models are used as further degrees of freedom to optimize the volume of the provable stability domain.

In future work, the approach will not only be used for a stability analysis of dynamic systems but also to optimize controllers so that the domains of attraction of stable operating points become as large as possible. Moreover, it is reasonable to consider not only set-valued uncertainty representations, but also links to techniques which simultaneously allow for robustifying control procedures in the presence of stochastic noise [11, 28, 32]. Finally, applications to continuous-time processes, in combination with the new invariance test sketched in Sec. 4, will be investigated.

References

[1] Althoff, M. and Rath, J. Comparison of guaranteed state estimators for linear time-invariant systems. *Automatica*, 130:109662, 2021. DOI: 10.1016/j.automatica.2021.109662.

[2] Barmish, B.R. *New Tools for Robustness of Linear Systems.* Macmillan, New York, 1994.

[3] Bourgois, A. *Safe & Collaborative Autonomous Underwater Docking.* PhD thesis, ENSTA Bretagne, Brest, France, 2021.

[4] Bourgois, A. and Jaulin, L. Interval centred form for proving stability of non-linear discrete-time systems. In Dang, Thao and Ratschan, Stefan, editors, *Proceedings 6th International Workshop on Symbolic-Numeric methods for Reasoning about CPS and IoT*, Volume 331 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–17. Open Publishing Association, 2021. DOI: 10.4204/EPTCS.331.1.

[5] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, 1994. DOI: 10.1137/1.9781611970777.

[6] Chapoutot, A. Interval slopes as a numerical abstract domain for floating-point variables. *Lecture Notes in Computer Science*, pages 184—200, 2010. DOI: 10.1007/978-3-642-15769-1_12.

[7] Chesi, G. On the estimation of the domain of attraction for uncertain polynomial systems via LMIs. In *43rd IEEE Conference on Decision and Control*, Volume 1, pages 881–886, 2004. DOI: 10.1109/CDC.2004.1428796.

[8] Chesi, G. Robust static output feedback controllers via robust stabilizability functions. *IEEE Transactions on Automatic Control*, 59(6):1618–1623, 2014. DOI: 10.1109/TAC.2013.2293453.

[9] Chilali, M. and Gahinet, P. H_∞ design with pole placement constraints: An LMI approach. *IEEE Transactions on Automatic Control*, 41(3):358–367, 1996. DOI: 10.1109/9.486637.

[10] Cornelius, H. and Lohner, R. Computing the range of values of real functions with accuracy higher than second order. *Computing*, 33:331–347, 1984. DOI: 10.1007/BF02242276.

[11] Dehnert, R., Damaszek, M., Lerch, S., Rauh, A., and Tibken, B. Robust feedback control for discrete-time systems based on iterative LMIs with polytopic uncertainty representations subject to stochastic noise. *Frontiers in Control Engineering*, 2, 2022. DOI: 10.3389/fcteg.2021.786152.

[12] Delanoue, N., Jaulin, L., and Cottenceau, B. An algorithm for computing a neighborhood included in the attraction domain of an asymptotically stable point. *Communications in Nonlinear Science and Numerical Simulation*, 21(1):181–189, 2015. DOI: 10.1016/j.cnsns.2014.08.034.

[13] Goubault, E. and Putot, S. Robust under-approximations and application to reachability of non-linear control systems with disturbances. *IEEE Control Systems Letters*, 4(4):928–933, 2020. DOI: 10.1109/LCSYS.2020.2997261.

[14] Goubault, E. and Putot, S. Tractable higher-order under-approximating AE extensions for non-linear systems. In *Proceedings of the 7th IFAC Conference on Analysis and Design of Hybrid Systems*, 2021. DOI: 10.1016/j.ifacol.2021.08.504.

[15] Hachicho, O. and Tibken, B. Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Volume 3, pages 3150–3155, 2002. DOI: 10.1109/CDC.2002.1184354.

[16] Jaulin, L., Kieffer, M., Didrit, O., and Walter, É. *Applied Interval Analysis*. Springer–Verlag, London, 2001. DOI: 10.1007/978-1-4471-0249-6.

[17] Kersten, J., Rauh, A., and Aschemann, H. Interval methods for robust gain scheduling controllers: An LMI-based approach. *Granular Computing*, pages 203–216, 2020. DOI: 10.1007/s41066-018-00147-1.

[18] Khalil, H.K. *Nonlinear Systems*. Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 2002.

[19] Marquez, H.J. *Nonlinear Control Systems*. John Wiley & Sons, Inc., New Jersey, 2003.

[20] Mayer, G. *Interval Analysis and Automatic Result Verification*. De Gruyter Studies in Mathematics. De Gruyter, Berlin/Boston, 2017. DOI: 10.1515/9783110499469.

[21] Merlet, J.-P. Interval analysis for certified numerical solution of problems in robotics. *International Journal of Applied Mathematics and Computer Science*, 19(3):399–412, 2009. DOI: 10.2478/v10006-009-0033-3.

[22] Monfared, M.N. and Yazdanpanah, M.J. Optimal dynamic Lyapunov function and the largest estimation of domain of attraction. *IFAC-PapersOnLine, Proceedings of the 20th IFAC World Congress*, 50(1):2645–2650, 2017. DOI: 10.1016/j.ifacol.2017.08.469.

[23] Nedialkov, N.S. Interval tools for ODEs and DAEs. In *CD-Proceedings of the 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics*, Duisburg, Germany, 2007. IEEE Computer Society. DOI: 10.1109/SCAN.2006.28.

[24] Pepy, R., Kieffer, M., and Walter, E. Reliable robust path planning with application to mobile robots. *International Journal of Applied Mathematics and Computer Science*, 19(3):413–424, 2009. DOI: 10.2478/v10006-009-0034-2.

[25] Pursche, T., Swiatlak, R., and Tibken, B. Estimation of the domain of attraction for nonlinear autonomous systems using a Bezoutian approach. In *SICE International Symposium on Control Systems*, pages 1–6, 2016. DOI: 10.1109/SICEISCS.2016.7470159.

[26] Rauh, A., Bourgois, A., and Jaulin, L. Union and intersection operators for thick ellipsoid state enclosures: Application to bounded-error discrete-time state observer design. *Algorithms*, 14(3):88, 2021. DOI: 10.3390/a14030088.

[27] Rauh, A., Bourgois, A., Jaulin, L., and Kersten, J. Ellipsoidal enclosure techniques for a verified simulation of initial value problems for ordinary differential equations. In *Proceedings of the 5th International Conference on Control, Automation and Diagnosis (ICCAD'21)*, Grenoble, France, 2021. DOI: 10.1109/ICCAD52417.2021.9638755.

[28] Rauh, A., Dehnert, R., Romig, S., Lerch, S., and Tibken, B. Iterative solution of linear matrix inequalities for the combined control and observer design of systems with polytopic parameter uncertainty and stochastic noise. *Algorithms*, 14:205, 2021. DOI: 10.3390/a14070205.

[29] Rauh, A. and Jaulin, L. A computationally inexpensive algorithm for determining outer and inner enclosures of nonlinear mappings of ellipsoidal domains. *International Journal of Applied Mathematics and Computer Science*, 31(3):399–415, 2021. DOI: 10.34768/amcs-2021-0027.

[30] Rauh, A. and Jaulin, L. A novel thick ellipsoid approach for verified outer and inner state enclosures of discrete-time dynamic systems. In *Proceedings of the 19th IFAC Symposium System Identification: Learning Models for Decision and Control*, Padova, Italy (online), 2021. DOI: 10.1016/j.ifacol.2021.08.426.

[31] Rauh, A., Kersten, J., and Aschemann, H. Toward the optimal parameterization of interval-based variable-structure state estimation procedures. *Reliable Computing*, 25:118–132, 2017. www.cs.utep.edu/interval-comp/reliable-computing-25-pp-118-132.pdf.

[32] Rauh, A. and Romig, S. Linear matrix inequalities for an iterative solution of robust output feedback control of systems with bounded and stochastic uncertainty. *Sensors*, 21(9):3285, 2021. DOI: 10.3390/s21093285.

[33] Rauh, A., Senkel, L., and Aschemann, H. Interval-based sliding mode control design for solid oxide fuel cells with state and actuator constraints. *IEEE Transactions on Industrial Electronics*, 62(8):5208–5217, 2015. DOI: 10.1109/TIE.2015.2404811.

[34] Rauh, A., Senkel, L., and Aschemann, H. Reliable sliding mode approaches for the temperature control of solid oxide fuel cells with input and input rate constraints. In *Proceedings of the 1st IFAC Conference on Modelling, Identification and Control of Nonlinear Systems*, St. Petersburg, Russia, 2015. DOI: 10.1016/j.ifacol.2015.09.217.

[35] Rauh, A., Senkel, L., Kersten, J., and Aschemann, H. Reliable control of high-temperature fuel cell systems using interval-based sliding mode techniques. *IMA Journal of Mathematical Control and Information*, 33:457–484, 2016. DOI: 10.1093/imamci/dnu051.

[36] Rohn, J. Positive definiteness and stability of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 15(1):175–184, 1994. DOI: 10.1137/S0895479891219216.

[37] Romig, S., Jaulin, L., and Rauh, A. Using interval analysis to compute the invariant set of a nonlinear closed-loop control system. *Algorithms*, 12(12):262, 2019. DOI: 10.3390/a12120262.

[38] Rump, S.M. Expansion and estimation of the range of nonlinear functions. *Mathematics of Computation*, 65(216):1503–1512, 1996. DOI: 10.1090/S0025-5718-96-00773-9.

[39] Rump, S.M. IntLab — INTerval LABoratory. In Csendes, T., editor, *Developments in Reliable Computing*, pages 77–104. Kluver Academic Publishers, 1999. DOI: 10.1007/978-94-017-1247-7_7.

[40] Scherer, C. and Weiland, S. Linear matrix inequalities in control. In Levine, W.S., editor, *Control System Advanced Methods*, The Electrical Engineering Handbook Series, pages 24–1—24–30. CRC Press, Boca Raton, 2nd edition, 2011. DOI: 10.1201/b10384.

[41] Skelton, R.E. Linear matrix inequality techniques in optimal control. In Baillieul, John and Samad, Tariq, editors, *Encyclopedia of Systems and Control*, pages 1–10. Springer London, London, 2020. DOI: 10.1007/978-1-4471-5102-9_207-2.

[42] Stengel, R. *Optimal Control and Estimation*. Dover Publications, Inc., New York, USA, 1994.

[43] Swiatlak, R., Tibken, B., Paradowski, T., and Dehnert, R. Determination of the optimal quadratic Lyapunov function for nonlinear autonomous systems via interval arithmetic. In *European Control Conference (ECC)*, pages 297–303, 2015. DOI: 10.1109/ECC.2015.7330560.

[44] Tibken, B. Estimation of the domain of attraction for polynomial systems via LMIs. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Volume 4, pages 3860–3864, 2000. DOI: 10.1109/CDC.2000.912314.

[45] Tibken, B. and Hachicho, O. Estimation of the domain of attraction for polynomial systems using multidimensional grids. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Volume 4, pages 3870–3874, 2000. DOI: 10.1109/CDC.2000.912316.

[46] Valmórbida, G., Tarbouriech, S., and Garcia, G. Region of attraction estimates for polynomial systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 5947–5952, 2009. DOI: 10.1109/CDC.2009.5399969.

[47] Weinmann, A. *Uncertain Models and Robust Control*. Springer–Verlag, Wien, 1991. DOI: 10.1007/978-3-7091-6711-3.

Received 26th August 2021

Contents

Regular papers

ISSN 0324—721 X (Print)
ISSN 2676—993 X (Online)

Editor-in-Chief: Tibor Csendes