LIGHT-WEIGHT ACTIVE SECURITY SOLUTIONS FOR RESOURCE-CONSTRAINED ICPS

A thesis submitted to Auckland University of Technology in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Supervisor

Associate Prof. Roopak Sinha

Dr. Matthew Kuo

20 November 2023

By

Farzana Zahid

Engineering, Computer and Mathematical Sciences

Abstract

Industrial Cyber-Physical Systems (ICPS) are driving the 4th Industrial Revolution, significantly impacting the productivity and efficiency of all sectors, including industrial automation. Central to this revolution is the networking and digitisation of multi-domain and large-scale physical systems within the industrial context. However, the seamless convergence of the digital and physical world has made ICPS vulnerable to new and sophisticated security threats. Ensuring the security requirements of ICPS is paramount, especially against cyber attacks that can significantly impact the availability of critical ICPS applications. The ICPS applications traditionally execute on resource-constrained devices like PLCs. These devices have limited resources, and standard security measures are inadequate to safeguard them due to the resource limitations. Balancing security requirements with distinctive characteristics of resource-constrained ICPS is pivotal for maintaining the performance, availability, and robustness of ICPS applications.

In this research, the significant contributions of our works are framed as research objectives and achieved using design science research methodology. We have presented several novel light-weight active security solutions developed to address the current gaps against cyber attacks, specifically Distributed Denial of Service (DDoS) attacks on the resource-constrained ICPS. DDoS attacks are the most reported attacks that disrupt or degrade the availability of systems, either by overloading them with a flood of packets or exploiting vulnerabilities. Considering

the disruptive and degrading impacts of DDoS attacks on the normal operations of the resource-constrained ICPS, this thesis focuses on detecting such attacks using light-weight active security solutions. The light-weight active security solutions are considered generic and programmable security measures that can proactively protect the devices with minimal overhead on their performance.

Systematic Mapping Study (SMS) and Systematic Literature Review (SLR) results have shown that light-weight active security solutions are crucial for resource-constrained ICPS to deal with DDoS attacks. We have also proposed the generic active security technique for detecting DDoS attacks on resource-constrained ICPS. Moreover, the notable inclusion of a novel multi-vector and cross-domain DDoS attack taxonomy helps us to devise the solutions for multi-scale flooding attacks and attack volumes and binary and multi-class slow-rate attack detection frameworks. The proposed works' effectiveness was determined using PLCs and publicly available datasets. The evaluations show noteworthy accuracy, low prediction time, and distinguished performance over existing state-of-the-art mechanisms.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

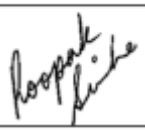
Publications

Table 1: List of Published/Submitted Research Articles

Manuscript	Publications/Submission	Contributions
1	Zahid, F., Kuo, M. M., & Sinha, R. (2021, December). Light-weight active security for detecting ddos attacks in containerised ICPS. In 2021 18th International Conference on Privacy, Security and Trust (PST) (pp. 1-5). IEEE. (Published)	Farzana Zahid: 90% Matthew Kuo: 5% Roopak Sinha: 5%
2	Zahid, F., Tanveer, A., Kuo, M. M., & Sinha, R. (2022). A systematic mapping of semi-formal and formal methods in requirements engineering of industrial cyber-physical systems. Journal of Intelligent Manufacturing, 33(6), 1603-1638. (Published)	Farzana Zahid: 80% Awais Tanveer: 10% Matthew Kuo: 5% Roopak Sinha: 5%
3	Zahid, F., Funchal, G., Melo, V., Kuo, M. M., Leitao, P., & Sinha, R. (2022, July). DDoS Attacks on Smart Manufacturing Systems: A Cross-Domain Taxonomy and Attack Vectors. In 2022 IEEE 20th International Conference on Industrial Informatics (INDIN) (pp. 214-219). IEEE. (Published)	Farzana Zahid: 80% Gustavo Funchal: 4% Victória Melo:4% Matthew Kuo: 4% Paulo Leitao: 4% Roopak Sinha: 4%
4	Funchal, G., Zahid, F., Melo, V., Kuo, M. M., Pedrosa, T., Sinha, R., & Leitao, P. (2023, April). An Intrusion Detection System Dataset for a Multi-Agent Cyber-Physical Conveyor System. In 2023 IEEE International Conference on Industrial Technology (ICIT) (pp. 1-6). IEEE. (Published)	Gustavo Funchal: 65% Farzana Zahid: 10% Victória Melo: 5% Matthew Kuo: 4% Tiago Pedrosa: 4% Roopak Sinha: 4% Fernando De la Prieta: 4% Paulo Leitao 4%
5	Zahid, F., Kuo, M. M., & Sinha, R. (2023). Actively Detecting Multi-Scale Flooding Attacks & Attack Volumes in Resource-Constrained ICPS. IEEE Transactions on Industrial Informatics (Manuscript No TII-23-3448, Re-Submitted after revision)	Farzana Zahid: 80% Matthew Kuo: 4% Roopak Sinha: 4% Gustavo Funchal: 4% Victória Melo:4% Paulo Leitao: 4%
6	Zahid, F., Kuo, M. M., & Sinha, R. (2023). Light-weight Slow-Rate Attack Detection Framework for Resource-Constrained Industrial Cyber-Physical Systems. International Journal of Information Security (Submission ID 1281cb87-9e48-445a-842e-dd6e33b473eb Under review).	Farzana Zahid: 90% Matthew Kuo: 5% Roopak Sinha: 5%

We, the undersigned, hereby agree to the percentages to the chapters identified above.

Table 2: Signatures of the Contributors

Name	Signature
Farzana Zahid	
Roopak Sinha	
Matthew Kuo	
Tanveer Awais	
Funchal Gustavo	Gustavo Silva Funchal Digitally signed by Gustavo Silva Funchal Date: 2023.11.16 10:36:55 Z
Victória Melo	Victória Clarissa de Abreu Melo Assinado de forma digital por Victória Clarissa de Abreu Melo Dados: 2023.11.16 10:44:34 Z
Paulo Leitão	Assinado por: PAULO JORGE PINTO LEITÃO Num. de Identificação: 08446321 Data: 2023.11.15 21:25:34+00'00'
Tiago Pedrosa	Assinado por: Tiago Miguel Ferreira Guimarães Pedrosa Data: 2023.11.16 11:36:37 +0000
Fernando De la Prieta	DE LA PRIETA PINTADO FERNANDO - 71011007W Firmado digitalmente por DE LA PRIETA PINTADO FERNANDO - 71011007W Fecha: 2023.11.17 08:08:34 +01'00'

Acknowledgements

First and foremost, I praise and thank God Almighty for his greatness and blessings and for giving me the opportunity, strength, and courage to complete this thesis. I would like to express my deepest gratitude to my primary supervisor, Roopak Sinha, for his support, guidance, and mentorship throughout the research. His constructive feedbacks were influential in enhancing the quality of this work. Thank you for everything!

I extend my deepest appreciation to my secondary supervisor, Matthew Kuo, for his invaluable inputs, prompt responses and thoughtful discussions. A special thanks to Barry Dowdeswell for offering your time whenever I needed it and for proofreading my thesis.

I am profoundly thankful to my beloved parents, whose efforts and prayers have paved the way for the position I find myself in today. This thesis is dedicated to my family, especially to my late father, whose memory serves as a source of inspiration. My heartfelt gratitude extends to my husband, and my lovely children. Their unwavering support, encouragement, and understanding during the ups and downs of this academic journey have allowed me to pursue and realise my dream.

Chapter 1

Introduction

Industrial Cyber-Physical Systems (ICPS) are recognised as the driving force behind Industry 4.0, and as a potential way to develop large-scale and complex systems faster and more effectively (Suvarna et al., 2021). ICPS is an integration of enormous physical devices, networking components, and computation in an industrial environment (Colombo et al., 2014). ICPS design and development include various disciplines, including control, mechanical, chemical, and software engineering. Due to the involvement of the cyber and physical worlds, ICPS feature heterogeneous devices, distributed stakeholders, and computers, distinguishing them from traditional embedded systems. Such interconnected systems evolve continually and contain emergent behaviours that arise from the seamless interaction and synchronisation between widespread and heterogeneous components and sub-systems (Zahid, Tanveer, Kuo & Sinha, 2021; Agrawal & Kumar, 2022).

On the one hand, ICPS offers significant opportunities in various sectors, including smart cities, real-time health care, smart manufacturing, intelligent transportation, cyber defence, water treatment, smart grids, and aerospace. These opportunities positively influence value-chain contributors like the environment,

society, devices, humans, and the economy. On the other hand, the development and deployment of ICPS systems and software are intrinsically complex and challenging due to larger scale, dispersed and evolutionary requirements, heterogeneity of physical and virtual components, swift creation of their virtual environment, security attacks, inherent resource-constrained nature, timeliness in secure communication, and newer levels of business-to-machine and business-human-machine-human interactions.

1.1 Rationale

Security is an ever-growing concern of ICPS (Kayan, Nunes, Rana, Burnap & Perera, 2021; Agrawal & Kumar, 2022; Ding, Han, Xiang, Ge & Zhang, 2018). ICPS applications have become prevalent in critical infrastructure and are increasingly connected to the Internet to perform and manage industrial tasks. Due to heterogeneous infrastructure, increased connectivity, and digitisation, ICPS are more vulnerable to continuously evolving cyber attacks, including DoS and its variations like DDoS (many-to-one attack) (A. Singh & Jain, 2018; Verma & Bharot, 2023; Yadav & Mishra, 2023; Ding et al., 2018; Zacchia Lun, DÍnnocenzo, Smarra, Malavolta & Di Benedetto, 2019). DoS/DDoS attacks are the pressing cybersecurity challenge to the availability of critical ICPS applications that traditionally execute on resource-constrained devices like PLCs that have limited computing power, memory, storage capacity, battery power, and bandwidth (Agrawal & Kumar, 2022; Verma, De Leon, Breslin & O'Shea, 2023; Sivamohan, Sridhar & Krishnaveni, 2023; Zahid et al., 2022a; Verma et al., 2023). DoS/DDoS attacks exploit the vulnerabilities in the applications or protocols (called Slow−Rate Attack (SRA)) or flood the network, services, or devices with a large number of network packets in a short time period. These attacks cause buffer overflow, delays,

crashes, or bandwidth depletion by degrading or disrupting the accessibility of some or all components of ICPS and prohibiting the distribution of data and/or control over the network (Zahid et al., 2022a). Any disruption or degradation to the availability of these systems and their normal operations can lead to prohibitive development costs, catastrophic consequences or detrimental impacts on performance. Therefore, the in-time detection of DoS/DDoS is imperative.

Resource-constrained systems lack security measures, have specialised communication protocols, and encounter cross-layers or cross-domain cyber attacks, where an attack on one layer can have a direct or indirect impact on other layers. Traditional or existing security measures such as firewalls, IDS, forensics, and honeypots are well-known mechanisms. Nonetheless, these mechanisms are resource-intensive, do not provide dynamic or programmable solutions, and are incompatible with resource-constrained devices' software and hardware configuration, resulting in network delays (Verma et al., 2023). These factors indicate that the existing general security solutions are not designed for resource-constrained ICPS or partially or fully impact the performances of such systems (Xiao, Xu, Jia, Ma & Qi, 2017). Therefore, to address the security challenges posed by resource-constrained devices, relying solely on generic security solutions is insufficient.

Managing the security challenges of ICPS with the unique characteristics of resource-constrained systems becomes crucial and challenging in maintaining the performance, availability, and robustness of ICPS applications (Ding et al., 2018). Nonetheless, researchers have less focused on the importance of this management (Zahid, Kuo & Sinha, 2021a; Gunjal, Patel, Alzhouri & Ebrahimi, 2023; Zahid, Tanveer et al., 2021; Verma & Bharot, 2023; Verma et al., 2023). Considering the security concerns associated with resource-constrained ICPS, there is a critical need for light-weight active security solutions. These active security mechanisms can provide general, programmable, and highly dynamic security solutions for

dealing with the evolving attacks on resource-constrained ICPS (Kayan et al., 2021; Zahid, Kuo & Sinha, 2021a; Verma & Bharot, 2023) without overburdening the limited resources of ICPS.

1.2 Research Methodology

Research methodology ensures the effectiveness of a research study by systematically designing a research program. Several research methodologies are available in the literature, including quantitative, qualitative, empirical research, or design science research methodology (DSRM) (A. Gupta & Gupta, 2022; John, Jan & Richard, 2017). Each research methodology has its own focus and goal. For example, qualitative research methodology enables the researchers to understand the subjects and their social and cultural environment through observations, interviews, questionnaires, and documentary sources. Quantitative research methodology quantifies the relationship between two or more groups through structured data collection and analysis. Empirical research involves data collection and analysis to test hypotheses and generate theories. A systematic procedure of creating innovative solutions to address practical problems or challenges aligns well with the Design science research methodology (DSRM). DSRM enables an iterative problem-solving process and primarily focuses on artifacts (processes, algorithms, techniques, frameworks, systems) creation to improve or solve real-world problems. Ultimately, selecting the appropriate research methodology depends upon the nature and type of knowledge to be generated by research.

This research aims to design, develop, and evaluate innovative solutions for providing light-weight active security to resource-constrained ICPS. We have adopted a methodological framework proposed in (Muntean & Militaru, 2022) to conduct research for this thesis. DSRM is the foundation of the adopted framework.

This framework provides a multi-phase process involving problem identification and motivation, defining research objectives, design and development, validation, and communication. Table. 1.1 presents the mapping of the framework's phases with the corresponding activities/steps and outputs (artifacts) developed during our research. We have structured our thesis into five phases. Phase 0 involves two steps: the first is to identify the problem and motivation for the thesis, and the second is to define the research objectives. Subsequent phases include steps of design and development, validation, and communication. The proposed framework allows us to cycle back to earlier activities as needed. We iterated back to the activity of defining research objectives to further advance our research (based on the insights gained from previous iterations) during each phase.

Table 1.1: Phases, Activities/Steps and Corresponding Outputs

Phase	**Activities**	**Output (Artifacts)**
Phase 0	A01:Problem Identification and Motivation A02: Establish Research Objectives	O01:Research objectives
Phase 1	A11:Surveying current research directions in RE of ICPS	O11:SMS; Chapter 2
Phase 2	A21:Surveying for active security approaches in RC-ICPS A22:Generic Framework designing and development for active security in RC-ICPS	O21:SLR; O22:Generic framework for resource-constrained ICPS; Chapter 4
Phase 3	A31:Designing a taxonomy for DDoS attacks in ICPS	O31:DDoS attack taxonomy for ICPS; Chapter 6
Phase 4	A41:Technique design and development for flooding attack detection in RC-ICPS	O41:Technique for flooding attacks; Chapter 8
Phase 5	A51:Design and development of framework for slow-rate attack detection in RC-ICPS	O51:Framework for slow-rate attacks; Chapter 10

1.2.1 Problem Identification, Motivation and Research Objectives

Recent industrial trends and developments and literature studies help us to identify that ICPS are an essential part of critical infrastructure. If these systems are subject to DDoS attacks, these systems become unavailable or unresponsive, which can cause substantial disruptions to industrial operations. Due to the inherent

resource-constrained nature, protecting resource-constrained ICPS from DDoS using traditional security measures is challenging and has not received significant research attention (Xiao et al., 2017; Verma et al., 2023).

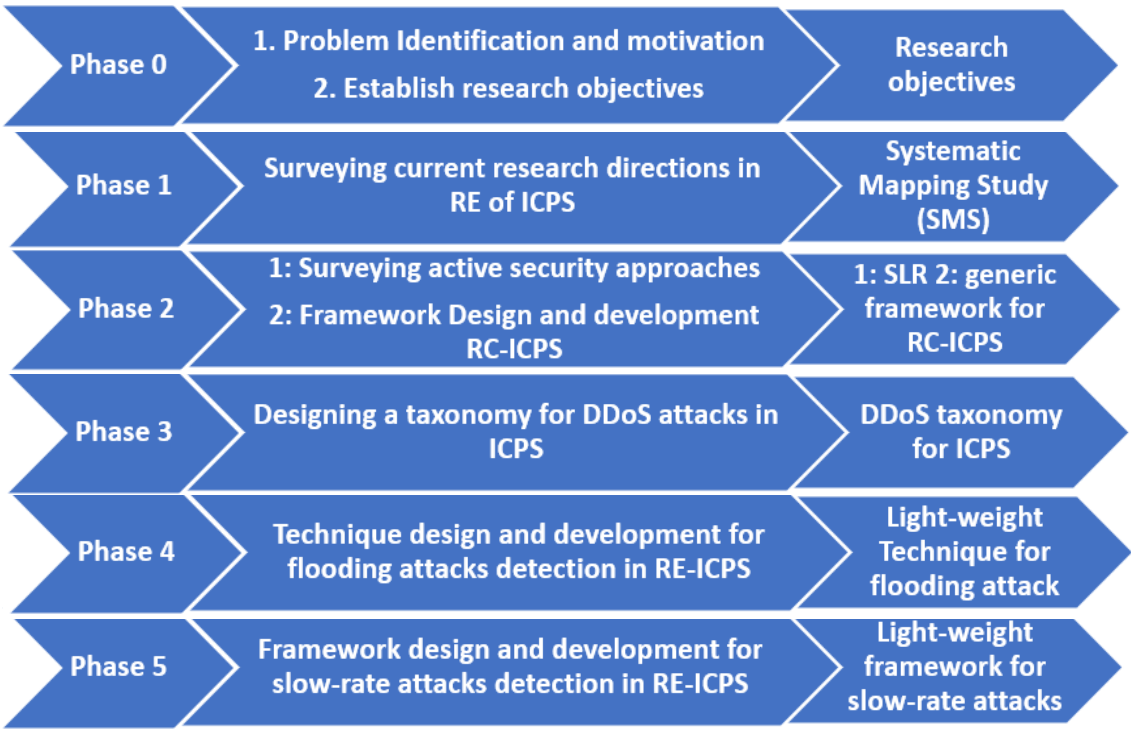


Figure 1.1: Research Methodology

The motivation of this thesis is to significantly enhance the self-protection of resource-constrained ICPS against evolving DDoS attacks using effective and efficient light-weight active security solutions. The research objectives (Figure. 1.1) of our research are outlined as follows:

1. To identify the challenges in ICPS, which needs further investigation. This objective is achieved through SMS, presented in Chapter 2.

2. To delve into a qualitative evaluation of key ICPS challenges, such as providing active security to detect DDoS attacks, specifically in the context of resource-constrained ICPS. SLR is used to investigate the state of evidence. This work is presented in Chapter 4.

3. To provide an overview of multi-vectors and cross-domain DoS/DDoS threat landscape within ICPS. This work is presented in Chapter 6.

4. To design and develop several light-weight active security solutions to address the identified challenges in actively securing resource-constrained ICPS. This objective is accomplished in three iterations, with corresponding works provided in Chapter 4, Chapter 8 and Chapter 10.

5. Evaluating the effectiveness of the proposed solutions using PLC and publicly available datasets.

1.2.2 Design and Development, Validation and Communication

Research objectives derived in phase 0 (Section. 1.2.1) are achieved iteratively through design and development, validation, and communication activities. The design and development activity in each phase is further divided into discrete sub-activities like systematic mapping study/ systematic literature review and artifacts engineering. In this thesis, the systematic mapping study is conducted to explore the current research directions within ICPS. The systematic literature reviews provide a comprehensive and qualitative evaluation of edge-cutting research studies for enhancing active security against DoS/DDoS attacks and their types (flooding and slow-rate attacks). The research findings and gaps help to create novel light-weight active security solutions (artifacts). The effectiveness of these solutions is demonstrated through simulations or case studies, evaluated to ensure how well the proposed solutions support the identified problem, and communicated through scholarly research publications at the end of each iteration.

In the first phase of design and development, a Systematic Mapping Study

(SMS) was conducted to identify the critical areas and trends within ICPS that require further attention and investigation. For the wider adoption, deployment, and successful operations of ICPS, the results of our SMS can be viewed from different perspectives, potentially shaping various research directions. Our SMS helps us to identify the techniques, tools, and frameworks that the researchers employed to enhance the security, performance, maintainability, and dependability of ICPS. The SMS also discusses various application domains, applicable industrial standards, and utilised formalism methods during different requirements engineering activities. The findings of the SMS contribute to the development of a novel conceptual model that highlights the trends in the domain. When considering further research, various research findings make it infeasible to focus on all areas. Hence, a researcher has to select an area of interest based on the knowledge, expertise, and significance of the selected domain. This approach empowers the researcher to contribute meaningfully and advance the understanding of critical aspects. Hence, the decision to focus on the security and performance aspects of ICPS serves as a focal point of this research. This choice is founded on the researcher's interest and expertise and the significant observations from our SMS regarding the often overlooked security and performance challenges in ICPS. ICPS applications are not incorporated with security by design and this opportunity opens the door for various vulnerabilities and threats to be exploited by attackers to launch cyber attacks. It has been found that the research community needs to give more attention to protecting against DoS/DDoS attacks that specifically impact the availability of ICPS applications. Publication 1 (Zahid, Tanveer et al., 2021) validates the findings of our SMS.

In phase 2, we first conducted the SLR of the existing generic and ICPS-related DoS/DDoS attack detection strategies in the literature (Chapter. 4).

The SLR leads to the identification of several research gaps such as resource-constrained ICPS are critical components in industrial environments; however, little work has been done related to resource-constrained ICPS. Due to resource constraints, they could not implement complex security measures; thus, a light-weight solution that can dynamically and programmability provide active security to resource-constrained ICPS applications is needed. Moreover, the majority of the research works focus on the analysis of the abnormal behaviour of a system within the time domain. The time domain analysis focuses on observing the network packets' timings as well as their contents and cannot separate harmonics, unusual spikes, and noise, which appear as random variations in traffic. This natural variation makes it challenging to identify the presence of DoS/DDoS attacks within the background network traffic. Instead of the time-domain representation, some light-weight and inexpensive domain logic is required to provide quick and easy identification of abnormal behaviour. To address the identified research gaps, we design, develop, and implement a novel general light-weight active security solution. Our proposed approach monitors and detects the presence of DoS/DDoS attacks on resource-constrained ICPS by creating and analysing the unique frequency signature of incoming network traffic (packets). Our approach employs a computationally efficient binning method for traffic pre-processing. In the binning method, the packets are grouped into contiguous and equal-width bins based on their timestamps. The timestamps represent the discrete events, and each bin has discrete packet counts. As the DoS/DDoS attacks exhibit specific temporal traits, binning methods help to identify the deviations or trends over different time intervals. It also ensures that traffic is processed appropriately for accurate analysis in later steps of the proposed technique. The bins of packet counts in the time domain are transformed into the frequency domain using a light-weight frequency transformation approach. This transformation provides an

easier and more accurate analysis of malicious behaviour or uncover the hidden patterns or trends (like noise and harmonics) in the traffic that are not detectable via time domain analysis. For attack detection, we first establish a baseline, an expected frequency signature for a system under normal operation. The frequency profile of the incoming traffic is continually compared to the baseline. If the current frequency pattern deviates significantly from the baseline, the system is considered to be under attack and generates an alarm. The experimental results based on different time-intervals and frequencies demonstrated that the proposed technique provides a light-weight mechanism for attack detection at a low computational cost. The proposed approach is executed as an algorithm in our proposed light-weight attack detection model designed for resource-constrained applications. Further details about the generic technique for providing active security in resource-constrained ICPS are found in Chapter 4. Furthermore, a conference publication (Zahid, Kuo & Sinha, 2021a) helped to evaluate our proposed solution through the research community.

The following research phase is to design and develop an easy-to-understand classification mechanism w.r.t cross-domain and analyse multi-vector DoS and DDoS attacks on ICPS (Chapter 6). ICPS offers flexible connectivity, interaction, and synchronisation among various physical and cyber components across different layers. The attack targeting one layer or domain can be propagated to other domains (cross-domain) or layers (cross-layer) that are not even attacked directly. For example, an attack on a controller in the cyber layer could lead to an erroneous control signal being propagated to the physical domain, causing unintended physical consequences. With the rapid evolution of DDoS attack types, multi-vectors, and duration, a significant challenge has emerged between maintaining product quality of service and protections against such attacks that can affect ICPS performance. The prevalence of multi-vector attacks is growing as attackers

integrate many DDoS techniques and tactics into a single composite attack (Sean, 2022). The cross-domain or layer and multi-vector attacks are sophisticated and need additional challenges for detection and prevention. In our proposed taxonomy, over fifty denial of service attacks on ICPS were classified as Endpoint and Network (distributed) Denial of Service attacks utilising the taxonomy. Endpoint (Distributed) Denial of Service attacks are launched for disrupting or degrading the availability of equipment or their services. An adversary conducts Network (Distributed) Denial of Service attacks by interrupting the communication among legitimate equipment or congesting/blocking the access of network resources (network bandwidth). Various sub-classes of Endpoint (distributed) Denial of Service attacks are also presented, including unavailability, delayed, buffer overflow, and manipulation attacks. Network (distributed) Denial of Service attacks are further classified as direct flooding, amplification, routing, and network jamming attacks. Thus, the taxonomy helps us identify various availability attacks in the manufacturing process that impact the availability of equipment, such as sensors, actuators, and controllers. Publication 3 (Zahid et al., 2022a) enabled this research to share the proposed taxonomy with the research community.

The subsequent iteration consolidates the extension of the proposed generic technique for DoS/DDoS detection (Chapter. 4). In this thesis, to achieve objective 4 (Section. 1.2.1), we propose a light-weight active security technique to detect multi-scale flooding attacks and volume in resource-constrained ICPS early from the incoming traffic flow (Chapter 8). The attack traffic exhibits multi-scale characteristics. The attacker sends the attack packets in predictable or random patterns with varying attack densities and at different time-intervals. Also, the attack volume (attack intensity) is a critical metric that adds a valuable dimension to the early detection of flooding attacks. It has been found from the existing literature that the optimal solution for detecting multi-scale flooding attacks and attack

volumes in resource-constrained ICPS is still an open research problem. Existing approaches are not memory-efficient, require significant processing capacity, and use a single method of detection that may have limited effectiveness in a changing environment. Also, the current works do not optimise performance overheads or acknowledge the attack volume. Our proposed technique initially pre-processes incoming traffic, transforming it into the frequency domain representation, as detailed in Chapter. 4. Subsequently, it employs a two-phase detection process to detect the presence of attacks and volumes. During the first phase, the presence of flooding attacks in incoming traffic is determined using a multi-method approach. The multi-method approach is a combination of light-weight and theoretically sound statistical methods. These methods detect an attack based on the incoming traffic's frequency signature deviation from the frequency profile of the baseline and the randomness of incoming traffic. The attack volumes are detected using a simple and computationally efficient dissimilarity metric. An alert will be triggered to provide details regarding the attack. In addition, to design and develop an efficient and accurate approach, the dynamic thresholds are also computed. The performance (memory and CPU overhead) of the proposed technique is evaluated using PLCs and publicly available datasets.

During the final iteration, our proposed taxonomy (Chapter. 6) played a significant role in addressing research objective 4 (Section. 1.2.1). Specifically, it helped in identifying that buffer overflow in ICPS is caused by SRA using less bandwidth, small computational resources and by generating less traffic (Tripathi & Hubballi, 2021; Reed, Dooley & Mostefaoui, 2021). To fulfil the fourth research objective, we propose an optimised OSELM-based, novel light-weight active security framework for binary and multi-class SRA detection in resource-constrained ICPS (Chapter. 10). Existing literature identifies that the research on detecting SRA in ICPS is still in its infancy, specifically in the

domain of PLC-based ICPS. The existing works are beyond the capacity of typical resource-constrained ICPS components because they either use a large number of features, have multi-layers with a large number of hidden nodes, or need to set the optimal hyperparameters, resulting in high processing and computation overhead and compromised significantly in dealing with diverse random traffic in a real-world environment. Also, less emphasis is given to minimizing the existing models' size, and more attention is required to maintain the performance overhead in the resource-constrained environment. Considering these issues, the aim is to enhance the performance of resource-constrained ICPS through the development of light-weight active security framework while maintaining optimal accuracy. The framework's foundation is an optimised OSELM model and a simple stratified-k fold training method. OSELM is an efficient online learning algorithm (G.-B. Huang, Liang, Rong, Saratchandran & Sundararajan, 2005). Our proposed framework further improves its resource efficiency by reducing its size and complexity. Additionally, the introduced training method reduces training duration and addresses the bias and overfitting issue of the imbalance dataset, enabling the optimised model to excel in online learning efficiently. The proposed framework is structured around data collection, training, and prediction components. The training component comprises data pre-processing and SRA detection training sub-components. The SRA detection training sub-component utilises a stratified-k fold training method to train and select the final optimised OSELM model. The model is trained to identify normal or attack traffic (binary detection) and to classify traffic into either normal or different slow-rate attack types, including Slowloris, Golden Eye, SlowHTTPTest, and Hulk (Rios, Inácio, Magoni & Freire, 2022). The prediction component includes the data pre-processing, SRA detection evaluation, and inference sub-components. During the data pre-processing, incoming traffic is analysed for any missing, undefined, or irrelevant information. It is then encoded

into a format the model can interpret, normalised to a common scale for an easier analysis, and randomly shuffled to improve the model performance. Moreover, a hybrid feature selection process is employed to determine the optimal features for model training. In the prediction phase, the model predicts using the pre-selected features established during the training phase to optimise the time and resource utilisation. In addition, the proposed framework has been evaluated on the publicly available dataset using PLC. Chapter 10 presents in-depth details about the proposed framework and obtained results.

The combination of the proposed artifacts (light-weight active security solutions) produced through each iteration of DSRM plays an essential role in the development of LISM for resource-constrained ICPS (Chapter. 12).

1.3 Thesis Organisation

This thesis presents in the "Thesis by manuscript" (Pathway 2) format, and the structure of the thesis is as follows.

Chapter 2 presents a detailed SMS and identifies the broader view of limitations in ICPS. The research objectives of the thesis are outlined in Chapters 4, 6, 8, 10. Each chapter focuses on the relevant research questions, details the background, literature reviews, proposed works, results, and conclusions. Chapter 12 discusses the overall research, highlighting significant findings, limitations, research's alignment with discipline and impact, threats to validity, conclusions, and future research directions, respectively. Appendix A presents a detailed description of the newly created intrusion detection dataset, which has been validated with our proposed solution discussed in Chapter. 8.

Chapter 2

Prelude - Manuscript 1

The following chapter was published as a journal article in the *Journal of Intelligent Manufacturing* under the title of *A systematic mapping of semi-formal and formal methods in requirements engineering of industrial cyber-physical systems* (Zahid, Tanveer et al., 2021). Manuscript 1 focuses on the first research objective (Section. 1.2.1), i.e., to identify the critical challenges and insights in ICPS, which needs further investigation.

The scale, heterogeneity, high complexity of ICPS, their evolutionary nature, and the involvement of multi-discipline stakeholders and machines have made Requirements Engineering (RE) of ICPS challenging. Ineffective RE can lead to incomplete and inaccurate requirements (both functional and non-functional), resulting in revenue loss, security compromises, and performance issues. Current RE methods (languages, techniques, frameworks, and tools) need to provide comprehensive and systematic support to deal with the unique challenges in ICPS. Thus, a well-defined approach to RE using Semi-Formal Methods (SFM) and Formal Methods (FM) in ICPS is essential (Fisher et al., 2014). Various solutions have been proposed in the literature and a holistic study of their relevance to RE in ICPS was currently missing.

This article used a multi-phase methodology called Systematic Mapping Study (SMS) (Petersen, Vakkalanka & Kuzniarz, 2015) and (BA & Charters, 2007) to provide a broader view for investigating and analysing the available formal and semi-formal methods in each activity of RE of ICPS. In this literature research, 3,645 papers were identified, out of which 93 primary studies were selected, using different inclusion and exclusion criteria and snowballing (searching through references and citations) techniques (Wohlin, 2014). We categorise the selected primary studies by quality (non-functional) requirements, application domains, research types and methods, applicable industrial standards, publication years, publication sources, and venue types. The findings of this study result in a novel conceptual model that highlights current trends in the area by showing the relationship between the RE activities of ICPS and formalisms. The conceptual model reflects the system engineering practices that could be helpful for both academic and industrial practitioners in selecting formal and semi-formal methods, illustration styles, and programming paradigms for their contexts.

We identify the critical gaps in the current literature, including the security areas that need further exploration to deal with cyber attacks, specifically Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks. These attacks significantly impact the availability of ICPS systems. Moreover, although resource-constrained ICPS are critical components for adequately functioning and controlling industrial processes, the researchers should have focused on this study area. Also, the performance of ICPS is overlooked in the literature. These research gaps, in particular, become the primary motivation for further research in this thesis.

Chapter 3

Systematic Mapping Study (Manuscript 1)

3.1 Abstract

The requirements engineering of Industrial Cyber-Physical Systems is extremely challenging due to large system sizes, component heterogeneity, involvement of multi-discipline stakeholders and machines, and continuous evolution. Formal and semi-formal languages, techniques, tools and frameworks can assist by providing repeatable and rigorous structures for eliciting, specifying, analysing, verifying and maintaining requirements. Various approaches have been proposed, but a contemporary and comprehensive study providing a landscape of the state-of-the-art is currently missing.

This article reports a systematic mapping study covering 93 primary studies published between 2009 and October 2020. We categorise surveyed studies by current research directions in the use of semi-formal and formal methods for Requirements Engineering phases for Industrial Cyber-Physical Systems. We also identify gaps in current research and develop a novel conceptual model capturing

the relationship between available formalisms and Requirements Engineering activities. We find that extensive work has been carried out on the formal analysis and verification of safety and timings requirements. However, the use of semi-formal notations, works on key phases like requirements elicitation and management, and the adoption of industrial standards are largely missing. Moreover, we find no literature providing methods to handle privacy and trust requirements, which have become critical concerns in this area.

3.2 Introduction

Cyber-Physical Systems are intelligent embedded systems that have tight coupling between their physical environment and computational components. ICPS are considered the driving force behind the fourth Industrial Revolution, where they promise a much-needed solution to the problem of developing increasingly complex and larger-scale systems faster. ICPS refer to the integration of large-scale physical processes, machines, computation, and networking components in an industrial environment (Colombo et al., 2014). ICPS span multiple disciplines, including chemical, mechanical, control and software engineering. Owing to the merging of virtual and physical worlds, ICPS feature geographically dispersed stakeholders, devices and computers, which differentiate them from traditional embedded systems. Such integrated systems also undergo continuous evolution and contain emergent behaviours that arise due to the long-term interaction between heterogeneous components and sub-systems. On the one hand, ICPS promise substantial opportunities in sectors like smart manufacturing, smart cities, intelligent transportation, real-time health care, smart grids, cyber defence, aerospace and water treatment, with positive impacts on value-chain contributors such as society, environment, humans, devices and the economy (Öztemel &

Gursev, 2020). On the other hand, evolutionary and dispersed requirements from a wide range of stakeholders, consideration for emergent behaviour, the scale and heterogeneity of collaborating physical and cyber components, incorporating new business and technological models, new levels of business-human-machine-human and business-machine interactions, amongst many others, have made developing ICPS systems and software intrinsically complex and challenging (Colombo et al., 2014).

The high complexity of ICPS requirements has a profound effect on all system engineering activities of the V-process model (Krueger, Walden & Hamelin, 2011). Hence, RE of ICPS, which covers requirements elicitation, analysis, specification, verification and validation (V&V), and management (Loucopoulos & Karakostas, 1995), is critical. Systematic RE allows for the ongoing interaction between *problem* domain requirements and *solution* domain requirements (Wiesner, Hauge & Thoben, 2015). Insufficient RE leads to inadequate or unclear requirements resulting in prohibitive development costs. In contrast to other contexts, RE in ICPS involves several unique challenges:

1. Organisational and social aspects of ICPS become just as important as technological concerns.

2. Requirements changes, originating from a large, dispersed and dynamic group of stakeholders, happen over long periods due to long-term system evolution.

3. Multi-site industrial processes (order, production, service delivery) feature their own methods, tools and models, which must be reused and incorporated into the overall ICPS.

4. Elicitation, V&V and management activities have a higher emphasis in

ICPS because of globalisation, heterogeneity of products, domains and services constituting the system, and the collaboration of multi-discipline, multi-culture and multi-site stakeholders.

5. Requirements originating from innovations in the supply-chain cycle, newer industry standards, traceability of standards-based requirements, compliance and resilience must be included.

Current RE methods (languages, techniques, frameworks and tools) do not provide comprehensive and systematic support to deal with the unique challenges in ICPS. In general, RE methods can be informal, semi-formal or formal. *Informal methods* involve natural language requirements, which feature high flexibility and expressiveness but require considerable human effort due to ambiguity. Subsequently, they are difficult to automate (Colombo et al., 2014). *Semi-Formal Methods (SFM)* encompass notations and languages, such as UML, which feature precise syntax and common vocabularies to reduce ambiguity. However, they still require some human effort for interpreting their semantics and therefore, complete automation remains difficult (Ahmed & Robinson, 2007). *Formal Methods (FM)* feature well-defined syntax and semantics and are used to enable the correctness by construction principle in safety-critical contexts. The principle aims to *"avoid introducing errors as far as possible, and remove those errors that are introduced, as soon as possible"* (Hall, 2005). However, formal methods require specialized expertise and training and do not scale as well as informal or semi-formal methods (Guttag et al., 1993).

Systematic RE in ICPS requires semi-formal and formal methods because informal methods cannot be used reliably for such large-scale and complex systems. SFM and FM are complementary and can be integrated. For instance, SFM may be used to engineer non-critical aspects of an ICPS such as handling customer

orders in manufacturing, while FM are better suited to handle safety-critical parts such as for functional safety in manufacturing (Hall, 2005). A well-defined approach to RE using SFM and FM in ICPS is essential (Fisher et al., 2014). While various piecemeal solutions have been proposed in the literature, a holistic study of how their relevance to RE in ICPS is currently missing.

This paper presents a SMS for the identification and analysis of existing semi-formal and formal methods that have been used in the requirements engineering of ICPS. Fig. 3.1 shows the scope of this study, which, to the best of our knowledge (evidenced by an analysis of published secondary works in Section 3.3) has not been covered by any existing secondary study. A total of 3,645 papers were identified

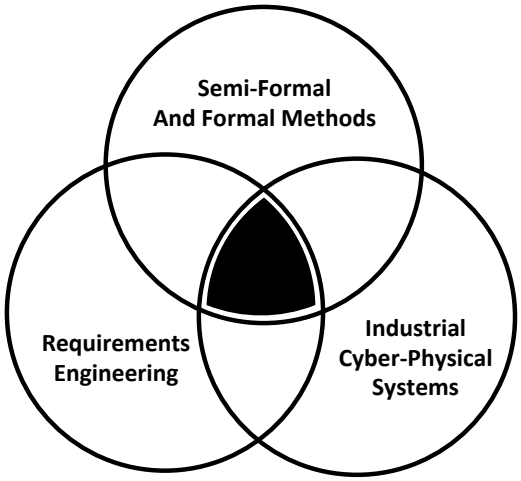


Figure 3.1: Scope for this systematic mapping study, restricted to studies publishes from 2009

out of which 93 primary studies were surveyed through the process described in Section 3.4. As presented in Section 3.5, we categorize the selected primary studies by quality (non-functional) requirements, application domains, research types and methods, applicable industrial standards, publication years, publication sources and venue types. A discussion on our findings appears in Section 3.6, which includes identifying critical gaps in current literature and a comparison

of the formalisms used in surveyed works. This leads to the development of a conceptual model that can aid academic and industrial practitioners to understand and choose appropriate semi-formal and formal methods for the requirements engineering of their ICPS. Concluding remarks and future directions are discussed in Section 3.7.

3.3 Related Works

A number of secondary studies that are somewhat related to our work are listed and compared in Table 3.1. These works were identified via the SMS process described later in Section 3.5.

Secondary studies such as (Zheng et al., 2015; Wiesner et al., 2015; Zheng & Julien, 2015; Penzenstadler & Eckhardt, 2012; Rashid et al., 2019; Simon et al., 2019) focus explicitly on cyber-physical systems, and others like (Yu et al., 2019; Wiesner et al., 2017; Colombo et al., 2017; X. Wu et al., 2020) relate directly to ICPS. However, none of these studies carries out a structured SMS, categorises research, discusses the categorisation of quality requirements except in (Simon et al., 2019) and considers formalisms except in (Zheng et al., 2015). (Zheng et al., 2015) perform a broad systematic literature review, a quantitative online survey involving 25 CPS researchers, and qualitative interviews with 9 CPS experts across four continents to uncover the existing approaches and practices for only formal verification and validation in CPS. Similarly, (Simon et al., 2019) provide a guideline for performing RE in small and medium-sized enterprises and define the general requirements to be met by a system. Unlike this work, our study focuses on the mapping of both formal and semi-formal methods to all the activities of RE and to identify the quality requirements catered by selected primary studies.

(Wiesner et al., 2015) elaborate on the challenges of RE, specifically for CPS

Table 3.1: Comparison of our work with existing secondary studies

References	Domains	RE Activities	Forma -lism	Identify quality requirements	Categorize Research	Types of Research	Model
(Simon, Felex & Jivka, 2019)	Cyber-Physical System/ SMEs	All RE activities	×	Yes	No	Opinion paper	No
(Penzenstadler & Eckhardt, 2012)	Cyber-Physical System/ SoS	Requirements Elicitation and Specification	×	No	No	Experi -ence paper	Yes
(Rashid, Siddique & Tahar, 2019)	Cyber-Physical system	Requirements Verification	×	No	No	Survey	No
(Wiesner et al., 2015)	Cyber-Physical system	Generic (RE process)	×	No	No	Survey	No
(Zheng & Julien, 2015)	Cyber-Physical system	×	×	No	No	Opinion paper	No
(Zheng, Julien, Kim & Khurshid, 2015)	Cyber-Physical System	Requirements Verification and Validation	FM	No	No	Survey	No
(Colombo, Karnouskos, Kaynak, Shi & Yin, 2017)	Industrial Cyber-Physical system	Generic	×	No	No	Opinion paper	Yes
(Wiesner, Marilungo & Thoben, 2017)	Cyber Physical Production System	Generic (RE Process)	×	No	No	Opinion paper	No
(X. Wu, Goepp & Siadat, 2020)	Cyber Physical Production system	Generic	×	No	No	Survey	Yes
(Yu, Dillon, Mostafa, Rahayu & Liu, 2019)	Industrial Cyber-Physical System	×	×	No	No	Experie -nce paper	Yes
(Lana, Guessi, Antonino, Rombach & Nakagawa, 2019)	Systems-of-Systems	Requirements Modelling	FM, SFM	No	No	SMS	Yes
(Vegendla, Duc, Gao & Sindre, 2018)	Software Ecosystems	All RE activities	×	Yes	No	SMS	No
(Hachicha, Halima & Kacem, 2019)	Self-adaptive System	Requirements Verification	FM	Yes	No	Survey	No
(Gabmeyer, Kaufmann, Seidl, Gogolla & Kappel, 2019)	Generic	Requirements Verification	FM	No	No	Survey	No
(Wortmann, Barais, Combemale & Wimmer, 2019)	Industry 4.0	Generic (Modelling notations)	Generic	No	No	SMS	Yes
(Takbiri & Amini, 2019)	Large-Scale Systems	Generic	×	No	No	Survey paper	No
(Sepúlveda, Cravero & Cachero, 2016)	Software Product Lines	Generic (Modelling Notations)	×	No	No	Survey	No
(Sharma & Singh, 2013)	Generic	Requirements Specification	FM	No	No	Survey	No
(Davis et al., 2013)	Generic	×	FM	No	No	Survey	No
(You, Li & Xia, 2012)	Generic	×	FM	No	No	Survey	No
Our Work	*Industrial Cyber-Physical System*	*All RE Activities*	*FM, SFM*	*Yes*	*Yes*	*SMS*	*Yes*

in distributed environments. (Zheng & Julien, 2015) present a study of developers on the topic of debugging of CPS. (Colombo et al., 2017) and (Yu et al., 2019) present a generic opinion and experience paper, respectively, to describe the fourth industrial revolution. They discuss the contribution, progress, world-wide emerging trends and challenges posed by ICPS. Likewise, (Wiesner et al., 2017), discuss the general challenges for RE process in cyber-physical production systems. (X. Wu et al., 2020) present a survey to examine the academic maturity of cyber-physical production systems. Our work extends these seminal works by providing a comprehensive landscape of how SFM and FM have been used in RE for ICPS with additional categorisations of the selected primary studies based on research type and quality requirements.

(Lana et al., 2019) perform a systematic mapping study of formal and semi-formal languages and techniques for requirements modelling of software-intensives systems-of-systems. (Penzenstadler & Eckhardt, 2012) propose a requirements engineering content model tailored for cyber-physical systems, to be used for requirements elicitation and specification. In contrast, our work focuses on all RE activities.

A mapping study of all the requirements engineering activities in software ecosystems is presented in (Vegendla et al., 2018). This study shows how quality requirements are considered in the software ecosystem, and unlike our work, it neither categorises surveyed works based on formalisms and research type nor presents a conceptual model integrating its overall findings.

Studies like (Hachicha et al., 2019; Gabmeyer et al., 2019) deal with only formal verification in RE. In the case of generic modelling notations, (Sepúlveda et al., 2016) discover 54 primary studies for software product lines. (Wortmann et al., 2019) identify 408 publications to assess the use of modelling languages in Industry 4.0. For large-scale systems, (Takbiri & Amini, 2019) analyse different studies

discussing large-scale requirements. None of these works focuses on requirements engineering related activities, quality requirements and formalisms.

A survey of formal requirements specification is presented in (Sharma & Singh, 2013). The paper evaluates specification languages such as object constraint, specification and description and Z languages. Unlike our work, they do not present any conceptual model and do not categorise their works on the quality requirements and research types.

A study by (You et al., 2012) surveys formal methods employed for the development of software. (Davis et al., 2013) survey the barriers faced by the USA government and private large systems manufacturers while adopting formal methods. They present their work generally instead of focusing on any particular domain. Furthermore, they do not provide categorisation by quality requirements, requirements engineering activities and research types.

Overall, this article provides a broader view for investigating and analysing the use of formal and semi-formal languages, techniques, tools and frameworks in each activity of RE in the context of ICPS. We identify the types of requirements targeted in the selected primary studies. Additionally, we contextualise our work to show the relationship between RE activities of ICPS and formalisms through an integrated conceptual model.

3.4 Systematic Mapping Study (SMS)

SMS is a multi-phase methodology consisting of planning, searching and reporting for identification, analysis and classification of existing literature in a particular domain, along with counting the contributions relating to the categories of that classification (BA & Charters, 2007). The major purpose of a mapping study is to identify areas of activity within a relatively larger scope, as compared to

systematic reviews. A systematic mapping, therefore, identifies the evidence base within the scope but does not delve into a qualitative evaluation. The SMS process involves the following steps: *planning*, *searching*, and *reporting*.

3.4.1 Mapping Study Protocol (Planning)

Planning contains the following steps.

Scope Definition: As previously shown in Fig. 3.1, the scope of this research is the intersection of three areas: semi-formal and formal methods, requirements engineering and industrial cyber-physical systems. In addition, we confine the search to works published in the last decade. This time-frame, between 2009 and October 2020, comprehensively covers the coining of the term ICPS in the early 2010s and the intense research activity in the area that followed in subsequent years.

Formulation of Research Questions: The scope leads to the following research questions:

- RQ1. What are the current research directions within the use of SFM and FM for the RE in ICPS?
  - RQ1.1. Which RE activities have been studied in the literature in the context of ICPS?
  - RQ1.2. Which SFM and FM (languages, techniques, tools and frameworks) have been utilized for performing the RE activities identified in RQ1.1?
  - RQ1.3. Which system's software requirements (functional and/or quality) of ICPS have been targeted by selected studies, discovered in RQ1.1 and RQ1.2?

- RQ2. What approaches are used, in literature, to assess the applicability of primary studies, identified in RQ1?
  - RQ2.1. Which application domains are used to determine the applicability of selected primary studies?
  - RQ2.2. Which research methods and research types have been employed in the selected primary studies?
- RQ3. Referring to RQ1 and RQ2, what are the observed state-of-art contributions of selected primary studies?
  - RQ3.1. Which industrial standards have been adopted in identified primary studies?
  - RQ3.2. How can the identified primary studies be classified according to the publication years?
  - RQ3.3. What are the publication sources and venue types for the identified primary studies?

Keywords Identification and Search String: Following several methods reported in
(Petersen et al., 2015), we identified the following keywords: `formal*`, `semi-formal`, `semi formal`, `requirement`, `specif*`, `valid*`, `verif*`, `elicit*`, `analy*`, `document*`, and `manag*`. Similarly, for ICPS, we used different combinations of words like `cyber`, `industrial`, `physical`, and `production`. These keywords were joined with `OR` and `AND` operators to form the search string shown below.

```
(ALL (formal*)  OR  ALL ("semi formal")  OR
ALL (semi-formal)  AND  ALL (requirement) AND
TITLE-ABS-KEY (elicit*)  OR
```

```
TITLE-ABS-KEY (analy*) OR
TITLE-ABS-KEY (specif*)  OR
TITLE-ABS-KEY (verif*)  OR
TITLE-ABS-KEY (valid*) OR
TITLE-ABS-KEY (document*)  OR
TITLE-ABS-KEY (manag*)  AND
ALL (industrial  AND cyber  AND
physical  AND system) OR
TITLE-ABS-KEY ("industrial cyber-physical
system")  OR  TITLE-ABS-KEY (cyber-physical
AND production AND system)  OR  TITLE-ABS-KEY
("cyber physical production system"))
AND  PUBYEAR  >  2008
```

It is important to note that the terms industrial cyber-physical systems and cyber-physical production systems are used interchangeably in literature.

Database Selection: (Franceschini, Maisano & Mastrogiacomo, 2016; Dyba, Dingsoyr & Hanssen, 2007) advocate choosing four or five robust databases relevant to the field of research. In this study, Scopus, IEEE Xplore, ACM Digital Library and SpringerLink databases were selected. Elsevier's Scopus is one of the largest commercially available database of peer-reviewed articles which also indexes IEEE Xplore, ACM Digital Library and SpringerLink. However, we searched all of these databases individually to find any articles that may not have been indexed by Scopus. The search string, shown above, is in the Scopus format and was tailored as needed for all other databases searched.

Inclusion and Exclusion Criteria: The following inclusion criteria (IC) and exclusion criteria (EXC) were used to select only the relevant studies from

the search results:

- IC1: Studies that investigated formal and/or semi-formal approaches for early- software development, i.e. at requirements engineering process of ICPS.
- IC2: Studies that focused only on research methods such as industrial case studies or industrial experimental work to assess and analyse their proposed solutions.
- IC3: Studies that were from computer science and software engineering or their sub-domains.
- IC4: Research studies that appeared since 2009 till October 2020.
- IC5: Studies published in conferences, journals, workshops, symposiums and technical reports, too.
- EXC1: Studies' whose title, keywords and/or abstract do not lie within the defined scope.
- EXC2: Studies that do not investigate any requirements engineering activity or FM/SFM related to ICPS.
- EXC3: Studies that address system-level or detailed design, unit testing, system integration testing, regression testing and acceptance testing.
- EXC4: Studies that do not provide an evaluation of proposed solutions.
- EXC5: Studies that do not include industrial applications.
- EXC6: Studies that address concepts such as networking/protocols, hardware, middleware, security requirements engineering, internet of things and cloud computing solely.

- EXC7: Studies that discuss challenges and problems in the targeted domain.
- EXC8: Studies that lack full text.
- EXC9: Books, thesis, secondary or tertiary studies, tutorial and opinion papers.
- EXC10: Studies not written in English.
- EXC11: Studies whose new version is available.
- EXC12: Duplicate articles found during search.

3.4.2 Searching

Search Execution: We used both automated and manual searching. *Automated systematic search* was conducted by following the protocol described in Section 3.4.1. The search execution chronology is shown in Fig. 3.2.

To find the potential primary studies, the initial search returned 3,645 articles using IC1 and IC3-IC5. Out of these, 40 articles were removed according to EXC11-EXC12 resulting in 3,605 studies. Secondly, a significant number of studies (1,478) were excluded by reading titles, keywords or abstracts (IC1 and EXC1) leaving 2,127 studies, and later by applying selection criteria IC2 and EXC6-EXC10, a further 1,000 studies were removed in order to select the related primary studies. The number dropped to 78 after a careful examination of introduction, conclusion and even full texts of each study (IC1 and EXC2-EXC5). In addition to automatic search, another searching technique called *snowballing* was also used to find the most relevant studies either through relevant references of each study or by using the citations related to already-included studies (Jalali & Wohlin, 2012). Forward or backward snowballing were used to identify and include another 15 relevant studies resulting in a total of 93 studies.

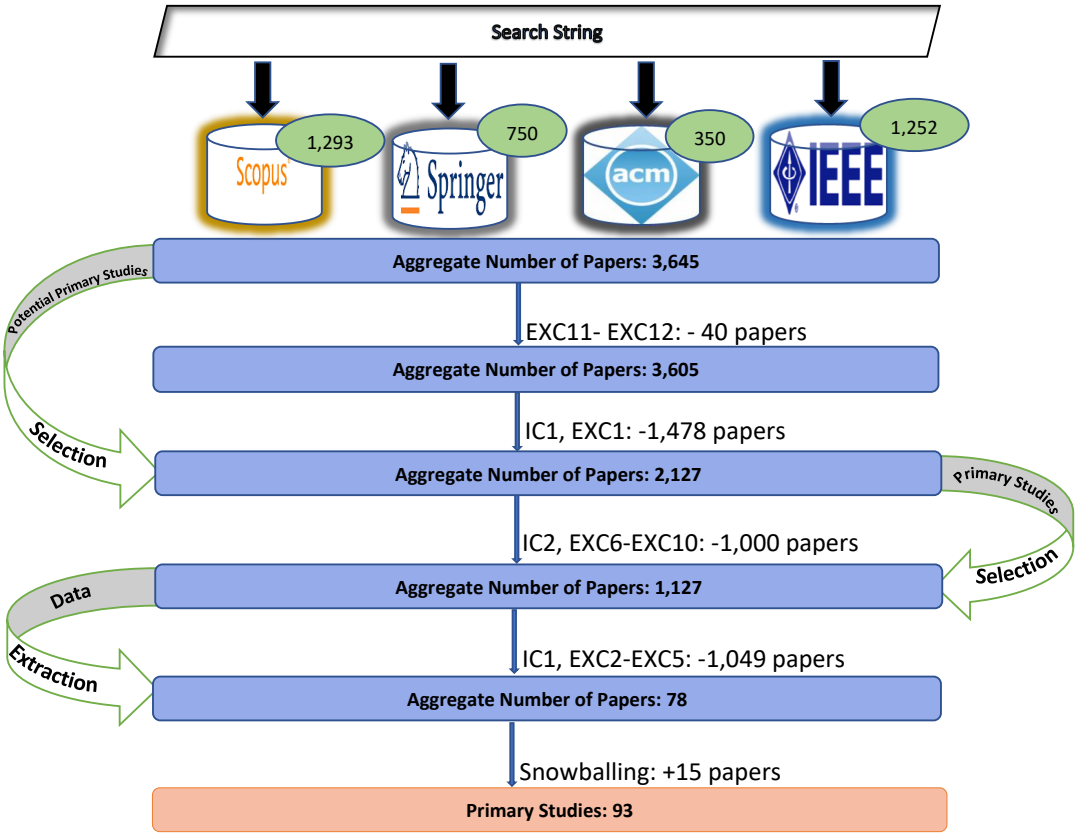


Figure 3.2: Search execution chronology

In manual searching, well-known sources such as the IEEE International Conference on Industrial Cyber Physical Systems, International Conference on Industrial Informatics, IEEE International Conferences on Software Testing, Verification and Validation Workshops, International Conference on Model Driven Engineering Languages and Systems, International Conference on Fundamental Approaches to Software Engineering, IEEE Transactions on Software Engineering, Springer's Requirements Engineering Journal, Springer's Formal Methods in System Design, IEEE International Conference of Requirements Engineering, IEEE International Workshop on Empirical Requirements Engineering were searched. Manual search helped reduce the probability of missing any relevant studies and

increase confidence in the depth of the review.

Data Extraction: Data extraction involves identifying keywords while reading abstracts together with the introduction and conclusion sections if the abstracts are not clear or well-written (Petersen et al., 2015). Open-coding enables the reviewer to allocate a 'code' or 'label' to terms or phrases, not necessarily according to their literal meaning but according to the concept behind the terms (Strauss & Corbin, 1998). This underpins the creation of a categorical and conceptual schema of the text or data in focus. Keywords identification followed the three-pass method described in (Keshav, 2007). According to this approach, deeper investigations are used to confirm the keywording, if an article's abstract, introduction or conclusion sections are ambiguous.

To address the research questions, we developed the rubric shown in Table 3.2, which employs both keywording and open-coding for data extraction. The data in Table 3.2 were analysed using text-mining techniques and maintained in MS-Excel to retrieve quality information, patterns and trends from the data set.

Table 3.2: Framework rubric for data extraction and classification

T1: Paper Title	T7. Publisher	T13. Languages used based on Formalism
T2. Authors	T8. Year of Publication	T14. Techniques, Frameworks and Tools
T3. Research Group/Organization	T9. Open Code and Keywords	T15. Types of Requirements
T4. Country	T10. Research Type	T16. RE Activities
T5. Pub. Venue	T11. Research Method	T17. Domains
T6. Pub. Type	T12. Formalisms	T18. Standards

(Felizardo, Nakagawa, Feitosa, Minghim & Maldonado, 2010) develop text mining tools, particularly for systematic mappings. Our data extraction rubric contains inherent features for implicit classification and categorisation. For tuple T9 of the rubric, the text-mining extension of Rapid Miner tool (Hofmann & Klinkenberg, 2013) was applied on open-coded keywords and phrases. The purpose of the tool was to identify patterns, trends and facets within the

research area. The outcomes of this analysis are discussed in detail in the following sections.

3.4.3 Conduction Reporting (Threats to Validity)

Missing Important Relevant Studies: SMS is intended to cover the breadth of a research area, unlike systematic literature review where the main objective is to analyse the current work in the field, regarding quality. Moreover, it is not possible to guarantee the coverage of all relevant existing literature. Thus, to overcome this threat, the search string is devised in such a way that it returns the maximum amount of studies from online databases. Sometimes, title and abstracts can also misdirect a reader if they are inarticulate. To mitigate this issue, we also read the introduction, conclusion and in some cases, the internal sections of a study to find any missing information from titles and abstracts. Furthermore, only specific databases were chosen, so there is a possibility of missing relevant studies. To alleviate this matter, a manual search of publication venues was also performed.

Researcher Bias: Researcher bias is another factor that may have affected the validity. To reduce the biases, the study mapping protocol was established carefully with the consent of domain experts and co-authors.

Number of Selected Relevant Studies: The size of the set of primary studies depends on the scope, novelty of research domains or clear separation between concepts. For example, in terms of RE activities, there is a vague line between concerned terms which can be used interchangeably by authors. Consequently, the primary studies were selected carefully to identify only relevant works by going through full text and case-studies of each study. Lastly, whenever there was a doubt, domain experts were consulted.

Researchers' expertise: Researchers' expertise is also a threat when considering

the research scope, shown in Fig. 3.1. Although, thorough knowledge of a particular area is mandatory, a researcher cannot be an expert in all aspects of a wide research area. This validity threat originates from the nature of systematic mapping process, which is wider and does not focus on qualitative aspects of the selected primary studies. To overcome this threat, researchers conferred with domain experts for feedback, advice, and exchange of ideas.

3.5 Results and Findings

We summarise the key results and findings of the SMS.[1]

3.5.1 Current Research Directions

Current research directions can be determined by answering the following sub-questions:

- RQ1.1 Which RE activities have been studied in the literature in the context of ICPS?
- RQ1.2 Which SFM and FM (languages, techniques, tools and frameworks) have been utilized for performing the RE activities, identified in RQ1.1?
- RQ1.3 Which system's software requirements (functional and/or quality) of ICPS have been targeted by selected studies, discovered in RQ1.1 and RQ1.2?

[1]All data is available for download via `www.removedforblindreview.com` in the form of an Excel worksheet.

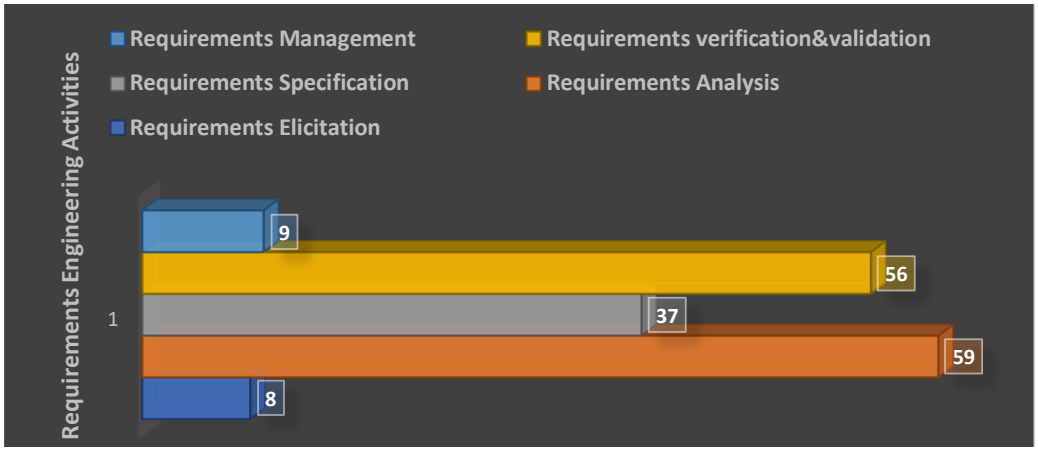


Figure 3.3: Classification of primary studies according to requirements engineering activities

RE activities in ICPS

The number of studies that stretch over different RE activities in ICPS is depicted in Fig. 11.1.

The figure shows that a single study can span over multiple activities, for instance (Mashkoor & Hasan, 2012; L. Zhang, He & Yu, 2013; Askarpour, Ghezzi, Mandrioli, Rossi & Tsigkanos, 2019) illustrate the use of FM on requirements analysis and V&V activities. A few target all RE activities, such as (Mancini et al., 2018). According to the results of mapping study, the most significant proportion of studies (59, 63%) cover requirements analysis followed by requirements V&V, covered by 56 (60%) studies. Requirements elicitation and management featured in only 8 (8.5%) and 9 (10%) studies, respectively.

The distribution of RE activities w.r.t publications and published years is depicted in Fig. 3.4. Overall, there is increasing attention in all activities of RE. V&V and analysis remain the dominant focus over the given period, except in 2009, where no work on requirements analysis was reported. By the end of the period (October 2020), only 5 research articles were found on requirements analysis while requirements V&V was featured in 7 studies. Studies on requirements elicitation

and management started appearing from 2014 but these areas remain relatively unexplored.

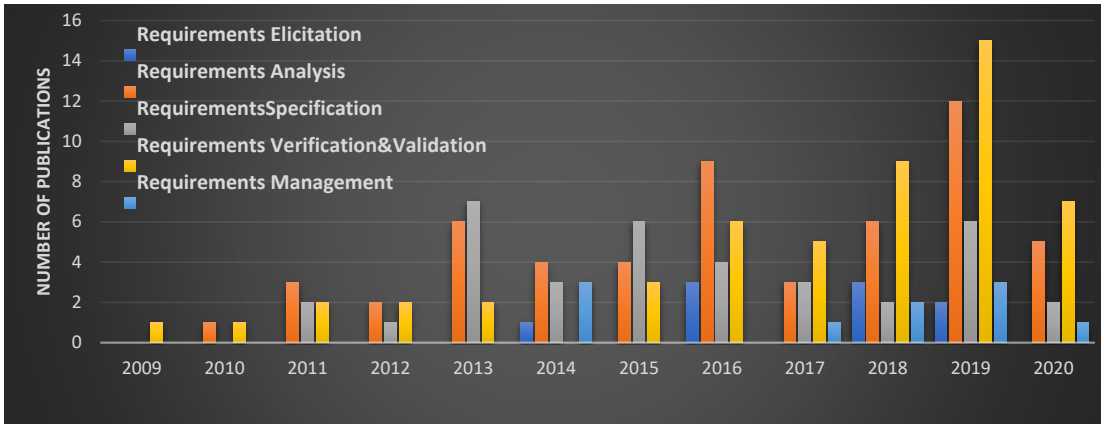


Figure 3.4: Requirements engineering activities w.r.t publication years

Semi-formal and Formal Methods in ICPS RE

First, we categorised the 93 primary studies based on formalisms such as formal methods, semi-formal methods or integration of both (Both). Next, the existing formal and semi-formal languages, techniques, frameworks or tools were plotted, solely, on identified RE activities (55%) while SFM are utilized in only 11 (11%) studies.

The classification of selected studies based on formalisms indicates that FM have been more the most studied concept. Out of 93 papers, 31 (33%) primary studies combined formal and semi-formal methods (both) in a single study. The numbers of primary studies that used FM are 51.

Requirements Elicitation: Requirements elicitation is a challenging task in ICPS. Table 3.3 shows that only 8 (10%) studies address this activity: (Denno & Blackburn, 2014; Seceleanu et al., 2017; Adepu & Mathur, 2016b; Nuzzo, Lora, Feldman & Sangiovanni-Vincentelli, 2018; Mühlfelder, 2018; Y. Chen, Dai, Zhang,

Pang & Vyatkin, 2018; J. Wang, Song, Wu & Dai, 2019; Loucopoulos, Kavakli & Chechina, 2019). In (Denno & Blackburn, 2014), the product data sheet is used

Table 3.3: Languages and techniques for requirements elicitation

Type	Method	Primary Studies
Language	Natural Language	(Denno & Blackburn, 2014), (Adepu & Mathur, 2016b), (Mühlfelder, 2018), (Nuzzo et al., 2018), (J. Wang et al., 2019), (Loucopoulos et al., 2019)
Techniques	Prototype	(Seceleanu et al., 2017)
	Graphical Notation	(Y. Chen et al., 2018)

to capture and communicate the equipment requirements among the designers and suppliers. Other studies use natural language to capture the requirements and used either prototypes or graphical notations for further clarifications. As an illustration, a wind turbine industrial prototype model is used in (Seceleanu et al., 2017) to clarify the timings requirement of the wind turbine system. A semi-formal natural language requirement description model is proposed in (J. Wang et al., 2019) for elicitation of requirements.

Requirements Analysis: Several formal and semi-formal languages, techniques, frameworks and tools have been used for performing requirements analysis activity.

A. Languages: Besides the activity of requirements specification, the reviewed primary studies used various formal and semi-formal specification languages during requirements analysis, too, as shown in Table 3.4.

These languages are: Architecture Analysis & Design Language (AADL) (Feiler & Gluch, 2012), Modelica (Elmqvist et al., 1999), Alloy, Unified Modelling Language (UML)(France, Evans, Lano & Rumpe, 1998), System Modelling Language (SysML) (Mann, 2009), Modeling and Analysis of Real-time and Embedded systems (MARTE) (Object Management Group, 2011), Parallel Object-Oriented Specification Language

(POOSL) (Theelen et al., 2007) and Process Algebra (PA)(Aceto, Ingólfsdóttir,

Table 3.4: Languages used in requirements analysis.

Languages	Variants	Primary Studies
AADL	-	(L. Zhang, 2013e), (L. Zhang, 2013b),(L. Zhang, 2013c),(L. Zhang, 2013d),(L. Zhang, 2014),(Ruchkin, 2015),(Akkaya, Derler, Emoto & Lee, 2016), (Lin, Adepu, Verwer & Mathur, 2018), (Zhan et al., 2019), (Misson, Gonçalves & Becker, 2019)
Modelica	-	(L. Zhang, 2013e), (L. Zhang, 2013b),(L. Zhang, 2013c), (L. Zhang, 2013d), (Gomez, Aguilera, Olsen & Vanfretti, 2020)
	Modelica ML	(L. Zhang, 2014)
Alloy		(Adepu, Kang, Jackson & Mathur, 2016)
POOSL		(Nägele, Broenink, Hooman & Broenink, 2019)
PA	ACP	(L. Zhang et al., 2013), (Mancini et al., 2018), (Sanwal & Hasan, 2013), (Adepu & Mathur, 2016c), (Goorden, van de Mortel-Fronczak, Reniers, Fokkink & Rooda, 2019), (Rashid & Hasan, 2020)
	SPA	(Akella & McMillin, 2009)
UML	-	(Mancini et al., 2018), (J. Wang et al., 2019), (Loucopoulos et al., 2019), (L. Zhang, 2013e), (L. Zhang, 2013c), (L. Zhang, 2014), (Kulvatunyou, Wallace, Ivezic & Lee, 2014), (F. Li, Zhang, Huang & Chen, 2016), (Iglesias et al., 2017), (Gomez et al., 2020), (Bernardi, Gentile, Marrone, Merseguer & Nardone, 2020)
	SysML	(L. Zhang, 2014), (Neghina, Zamfirescu & Pierce, 2019), (Vogel-Heuser, Schütz, Frank & Legat, 2014), (Pagliari, Mirandola & Trubiani, 2019), (Gomez et al., 2020), (Gräßler, Bodden, Pottebaum, Geismann & Roesmann, 2020)
	MARTE	(Seceleanu et al., 2017), (Ribeiro, Rettberg, Pereira & Soares, 2016), (L. Huang, Liang & Kang, 2019) , (D. Du, Huang, Jiang & Mallet, 2018)

Larsen & Srba, 2007) with its variants.

AADL is an architecture description language developed by SAE International. It is used to model hardware and software architecture of real-time embedded systems. Modelica is an object-oriented, equation-based programming language used to model complex physical system. As a result of the mapping process, we have found that AADL has been used in 10 studies. Among these, 5 selected studies (L. Zhang, 2013e, 2013b, 2013c, 2013d, 2014) integrate a Modelica with AADL, for specifying the cyber and physical part of automotive cyber physical systems, respectively. The reason for integration is that *"the descriptive power of AADL models with the analytic and computational power of Modelica models provides a capability that is significantly greater than provided by AADL or Modelica individually"*(L. Zhang, 2013e). Likewise, Alloy is a specification language used to describe the structure in a software system. Thus, in (Adepu et al., 2016), it is used to model connectors between components and behavioural aspect of the

secure water treatment model. Similarly, POOSL is a discrete-time modelling language and is used to developed embedded control software in (Nägele et al., 2019).

Process algebra (PA) deals with formal description or specification of concurrent processes. Algebra of Communicating Processes (ACP) and Security Process Algebra (SPA), belong to this family. ACP, along with its variants, are employed by six primary studies (L. Zhang et al., 2013; Sanwal & Hasan, 2013; Adepu & Mathur, 2016c; Goorden et al., 2019; Mancini et al., 2018; Rashid & Hasan, 2020) and SPA is employed only in (Akella & McMillin, 2009). ACP variants like second-order homogeneous linear differential equations are used in (L. Zhang et al., 2013) to model damped harmonic oscillation. An algebraic expression has been used in (Mancini et al., 2018) to split the requirements for the product line, and in (Sanwal & Hasan, 2013), process in-variants are used to detect attacks in industrial control systems. Similarly, ACP is used to determine the functional requirement of ICPS in (Adepu & Mathur, 2016c) and to analyse the smart grids in (Goorden et al., 2019). Moreover, in (Rashid & Hasan, 2020), ACP is utilized to formally analyse continuous dynamics of CPS, which are then verified by the theorem prover. In (Akella & McMillin, 2009), SPA is used to analyse security requirements of a natural gas transport system.

UML, SysML and MARTE are modelling languages used to model the system. SysML and MARTE are variants of UML. UML is used in eleven primary studies (Mancini et al., 2018; J. Wang et al., 2019; Loucopoulos et al., 2019; L. Zhang, 2013e, 2013c, 2014; Kulvatunyou et al., 2014; Bernardi et al., 2020; F. Li et al., 2016; Iglesias et al., 2017; Gomez et al., 2020). SysML is employed in (L. Zhang, 2014; Neghina et al., 2019; Vogel-Heuser et al., 2014; Pagliari et al., 2019; Gomez et al., 2020; Gräßler et al., 2020). In (L. Zhang, 2014), UML and SysML are integrated with Modelica ML (version of Modelica) to model the requirements of

cyber-physical systems of systems. Likewise, Modelica, UML and SysML have integrated in (Gomez et al., 2020) for the analysis of smart grid's functional and non-functional requirements. On the other side, four primary studies (Seceleanu et al., 2017; Ribeiro et al., 2016; L. Huang et al., 2019; D. Du et al., 2018) utilized MARTE along with its extensions.

B. Techniques: Formal methods based on the technique of Automata theory with their extensions are used by 12 primary studies (Seceleanu et al., 2017; von Birgelen & Niggemann, 2018; L. Zhang, 2013e; Kumar et al., 2012; Bu et al., 2011; Geraldes et al., 2018; Ye-Jing et al., 2013; Balasubramaniyan et al., 2016; R. Wang, Song, Zhu & Gu, 2011; Lin et al., 2018; D. Du et al., 2018; N. Li, Tsigkanos, Jin, Hu & Ghezzi, 2020), as presented in Table 3.5. The development

Table 3.5: Requirements analysis and formal techniques.

Techniques	**Variants**	**Primary Studies**
Automata	Hybrid Timed Automata	(Seceleanu et al., 2017),(Kumar et al., 2012),(Bu et al., 2011), (Geraldes et al., 2018),(Ye-Jing et al., 2013), (Balasubramaniyan et al., 2016), (R. Wang et al., 2011), (von Birgelen & Niggemann, 2018)
	Cellular Automata	(L. Zhang, 2013e)
	PDRTA	(Lin et al., 2018)
	Stochastic Timed Automata	(N. Li et al., 2020), (D. Du et al., 2018)
Graph Theory	Attack Execution Graph	(LeMay, Ford, Keefe, Sanders & Muehrcke, 2011)
		(Askarpour et al., 2019),(Adepu & Mathur, 2016a)
	Skill Graph	(Knüppel et al., 2020)
Abstract State Machine		(Metsälä et al., 2017), (Drozdov, Patil, Dubinin & Vyatkin, 2019)
Formal Ontology		(Y. Chen et al., 2018), (Kulvatunyou et al., 2014), (Sinha, Pang, Martínez, Kuronen & Vyatkin, 2015)
Formal Contract		(L. Zhang, 2013d), (L. Zhang, 2014),(Nuzzo et al., 2018), (Westman & Nyberg, 2014)
VDM	VDM-RT	(Neghina et al., 2019)

of modern-day computer science can be attributed to the work done using a theory of automata in the mid-20th century. A close relationship of automata theory with formal grammars makes it favourable to formally describe a model of a system as a state machine (Hopcroft, Motwani & Ullman, 2000). In eight primary studies (Seceleanu et al., 2017; von Birgelen & Niggemann, 2018; Kumar et al., 2012; Bu

et al., 2011; Geraldes et al., 2018; Ye-Jing et al., 2013; Balasubramaniyan et al., 2016; R. Wang et al., 2011), Hybrid Timed Automata (a version of automata) are used to model the behaviour of ICPS over time. While in (N. Li et al., 2020) and (D. Du et al., 2018), stochastic timed automata are used to represent timings and stochastic behavior in smart city and energy aware building. Cellular Automata is used by (L. Zhang, 2013e) for modelling and specification of the spatial-temporal requirements. Similarly, Probabilistic Deterministic Real-Time Automaton (PDRTA) is used with AADL to model the discrete events of the complex water treatment plant in (Lin et al., 2018).

Graph theory is another formal technique used in requirements analysis activity by (Askarpour et al., 2019), (LeMay et al., 2011), (Knüppel et al., 2020) and (Adepu & Mathur, 2016a) for ICPS modelling. It is closely related to automata and mathematical logic. In fact, the author in (Sakarovitch, 2009) defines a graph as a form of automata. For static modelling of topology for the smart city application domain, bigraphical technique is used in (Askarpour et al., 2019). An attack execution graph is used in a framework called ADversary VIew Security Evaluation (ADVISE) to find out the sequences, time, cost, probabilities of and other related information about each attack steps in (LeMay et al., 2011). Likewise, skill graphs in (Knüppel et al., 2020) are used to identify the poorly defined safety requirements at an early stage of hybrid system development.

Abstract state machines are used in (Metsälä et al., 2017; Drozdov et al., 2019) to capture states of ICPS for security analysis and to model distributed control systems based on the IEC 61499 production engineering standard, respectively.

Formal Ontology is an implicit or explicit conceptualization of axioms in formal language. Three primary studies (Y. Chen et al., 2018; Kulvatunyou et al., 2014; Sinha et al., 2015) used formal ontology for requirements analysis. The knowledge-base ontology is used in (Y. Chen et al., 2018) to formalise the

requirements as rules and relations in order to make it understandable for machines and humans. In addition to this, to describe the software components, functional ontology is identified in (Kulvatunyou et al., 2014) that is applied with use cases (UML diagram). A primary study by (Sinha et al., 2015) converts requirements specified in natural language into formalised ontologies for further analysis. The general idea is to extend initial requirements ontology during subsequent phases of V-process model.

Formal contracts are based on invariants, pre and post conditions of software functions. Four studies (L. Zhang, 2013d, 2014; Nuzzo et al., 2018; Westman & Nyberg, 2014) concentrate on contracts for the formalisation of the rules. The Analysis contracts framework in (L. Zhang, 2013d) is based on analysis-contracts that are used to analyse the security requirements for the water treatment system. In (Nuzzo et al., 2018), assume-guarantee (A/G) contracts are used to provide formal support to the high-level requirements in Contract-based Hierarchical Analysis and System Exploration (CHASE) framework. Additionally, (L. Zhang, 2014) and (Westman & Nyberg, 2014) defines the contracts for Simulink (Stateflow) diagrams and for structuring safety requirements in Fuel Level Display (FLD)-system respectively.

Vienna Development Method Real-Time (VDM-RT), used in (Neghina et al., 2019), is a formal method technique to model timings constraints of cyber physical production system. VDM-RT is a modified form of Vienna Development Method (VDM) that is one of the longest established formal method techniques to model industrial projects (J. Wang, 2007).

Semi-Formal Techniques are employed by eleven primary studies (J. Wang et al., 2019; Loucopoulos et al., 2019; Kulvatunyou et al., 2014; F. Li et al., 2016; Iglesias et al., 2017; Neghina et al., 2019; Vogel-Heuser et al., 2014; Ribeiro et al., 2016; Gomez et al., 2020; Gräßler et al., 2020; Bernardi et al., 2020), as depicted

in Table 3.6.

Table 3.6: Requirements analysis and semi-formal techniques

Technique	**Primary Studies**
Use-cases	(Kulvatunyou et al., 2014), (Gomez et al., 2020), (Bernardi et al., 2020)
Domain Model	(J. Wang et al., 2019), (Iglesias et al., 2017)
Meta-Models	(Loucopoulos et al., 2019), (F. Li et al., 2016), (Ribeiro et al., 2016), (Gräßler et al., 2020)
Requirements Diagram	(Neghina et al., 2019), (Gomez et al., 2020)
Parametric Diagram	(Vogel-Heuser et al., 2014)

The studies (J. Wang et al., 2019; Iglesias et al., 2017) define the domain model showing the monitoring of software family for ICPS and patient control system. Likewise, meta-models are used in (Loucopoulos et al., 2019; F. Li et al., 2016; Ribeiro et al., 2016; Gräßler et al., 2020). For example, in (F. Li et al., 2016), meta-model is used to show the relationship between service and functions blocks of industrial assembly line system. A requirement diagram is used to model the functional model unit of USB stick production line in (Neghina et al., 2019) and to support use cases that have been applied to determine the functional and non-functional requirements of the smart grid in (Gomez et al., 2020). SysML-AT (SySML extension), a specialized language profile to covers (non-) functional requirements, is adapted in (Vogel-Heuser et al., 2014) for automation, where SysML parametric diagrams models the components involved in the manufacturing process. It is noteworthy that diagrams used in semi-formal methods are considered as techniques in our work.

C. Frameworks: Table 3.7 shows the eleven primary studies that develop different domain-specific frameworks for requirements analysis. A formal attack model in (Adepu & Mathur, 2016b) is used to determine the cyber-attacks on a water treatment system. The study by (Loucopoulos et al., 2019) reports on the early Capability Oriented Requirements Engineering (e-CORE) framework for

analysis and traceability of requirements for automobile manufacturing industrial-size case study where the association between assets is shown by the meta-model.

A Secure Modelling Framework (SeMF) was developed to analyse the security

Table 3.7: Frameworks for requirements analysis

Frameworks	**Primary Studies**
Formal Attack Model	(Adepu & Mathur, 2016b)
Contract-based Hierarchical Analysis and System Exploration (CHASE)	(Nuzzo et al., 2018)
Early Capability Oriented RE (e-CORE)	(Loucopoulos et al., 2019)
Analysis Contract Framework	(Ruchkin, 2015)
Unified Graphical Framework	(Zhan et al., 2019)
ADversary VIew Security Evaluation (ADVISE)	(LeMay et al., 2011)
Secure Modelling Framework(SeMF)	(Fuchs, Gürgens, Weber, Bodenstedt & Ruland, 2010)
Real-Time Maude Framework	(Bae, Krisiloff, Meseguer & Ölveczky, 2015)
Clock	(B. Xu & Zhang, 2013)
SKEDITOR	(Knüppel et al., 2020)
Surreal	(Bernardi et al., 2020)

flaws in smart grid in (Fuchs et al., 2010). Likewise, a Unified Graphical Framework in (Zhan et al., 2019) is a graphical framework that is consists of AADL and Simulink (Stateflow) to model, simulate and validate ICPS. Real-Time Maude Framework in (Bae et al., 2015) is used to design a multirate distributed hybrid systems consisting of an airplane maneuvered by a pilot. In (B. Xu & Zhang, 2013), the clock theory concept is used to analyse the timing requirements of different applications like steam boiler control system, press and railway cross system. Similarly in (Knüppel et al., 2020), a framework called SKEDITOR is proposed that combined formalised skill graphs and theorem prover, KeYmaera X, to analyse and verify safety requirements in a domain of transportation. A framework known as Surreal is developed in (Bernardi et al., 2020) where security and safety requirements of smart cars are analysed by misuse cases and verified by a model checker (nuSMV).

D. Tools: Simulation and co-simulation are used for requirements analysis in 12 studies, as shown in Table 3.8. For simulation, Ptolemy II (Akkaya

Table 3.8: Simulation and co-simulation tools used in requirements analysis

Tools	Types	Primary Studies
Simulation	AADLSim	(Zhan et al., 2019)
	Simulink	(Sanwal & Hasan, 2013), (Lin et al., 2018), (Kang, Mu, Huang & Lan, 2018), (N. K. Singh & Wang, 2019), (Clarke & Zuliani, 2011)
	TrueTime	(Balasubramaniyan et al., 2016)
	PtolemyII	(Akkaya et al., 2016), (Pagliari et al., 2019)
Co-Simulation	20-sim	(Nägele et al., 2019)
	Overture Tool	(Neghina et al., 2019)
	FMI	(Gomez et al., 2020)

et al., 2016; Pagliari et al., 2019) models the performance of a delivery robotic system. Simulink is used in (Sanwal & Hasan, 2013; Lin et al., 2018; Kang, Mu et al., 2018; N. K. Singh & Wang, 2019; Clarke & Zuliani, 2011) for modelling transportation and manufacturing systems. Another simulation tool, AADLSim is used to model an Isollete system in (Zhan et al., 2019). TrueTime simulates performance requirementd of industrial mine pump in (Balasubramaniyan et al., 2016). Co-simulation can be performed either by tools or Functional mock-up Interface standard (FMI). FMI has been used in (Gomez et al., 2020) for multi-domain simulation whereas tools like 20-sim in (Nägele et al., 2019) and Overture in (Neghina et al., 2019) are used for co-simulation.

Requirements Specification: Table 3.9 shows the various formal and semi-formal specification languages have been used in the selected primary studies. Common Algebraic Specification Language (CASL) is a specification language that is based on first-order logic with mathematical proofs. Researchers have also used other variants of first-order logic. CASL-First Order Logic is employed in four studies. Eleven studies use the temporal extension of CASL called CASL-Temporal Logic (CASL-TL).

Four studies use Linear Temporal Logic (LTL). LTL can express sequence or paths (Huth & Ryan, 2004) of states of reactive systems over time. In (Clarke &

Zuliani, 2011), LTL's variant Bounded Linear Temporal Logic (BLTL) is used to express reliability properties of aircraft. Signal Temporal Logic (STL), another

Table 3.9: Formal and semi-formal requirements specification languages

Languages	**Variants**	**Primary Studies**
CASL-First Order Logic	Signal First Order	(Sanwal & Hasan, 2013),(Menghi, Nejati, Gaaloul & Briand, 2019), (Nejati et al., 2019),(Cengic & Akesson, 2010)
CASL-Temporal Logic (TL)	LTL	(Loucopoulos et al., 2019),(Kang, Huang & Mu, 2018),(Gawanmeh, Alwadi & Parvin, 2017), (Grobelna, 2020)
	LTL-BLTL	(Clarke & Zuliani, 2011)
	LTL-STL	(Nejati et al., 2019)
	LTL-MTSL	(Sun, Liu, Chen & Du, 2015)
	CTL	(Balasubramaniyan et al., 2016), (Meseguer & Ölveczky, 2012), (Gawanmeh et al., 2017),(Wisniewski, Grobelna & Karatkevich, 2020)
	CTL-TCTL	(Misson et al., 2019)
B Language	Event-B	(N. K. Singh & Wang, 2019)
Z Language		(Mashkoor & Hasan, 2012), (L. Zhang, 2011)
CSP	Timed CSP	(L. Zhang, 2013c),(L. Zhang, 2013a)
DSLs	Value Specification Language	(Ribeiro et al., 2016)
	ASL++ Language	(Rocchetto & Tippenhauer, 2017)
	RTCM	(Yue, Ali & Zhang, 2015)
	FORM-L	(Bouskela, Nguyen & Jardin, 2017)
State-Based Languages	SMV	(Drozdov, Patil & Vyatkin, 2017)
AADL		(Hissam, Chaki & Moreno, 2015), (E. Ahmad et al., 2015)
	AO4AADL	(L. Zhang, 2013a)
	EAST-AADL	(Kang, Huang & Mu, 2018), (Kang, Mu et al., 2018)
	Multirate Synchronous AADL	(Bae et al., 2015)

variant of LTL, is reported in (Nejati et al., 2019) to specify real-time temporal operators and real-valued constraints for smart manufacturing. Similarly, another LTL extension called Metric Temporal-Spatial Logic (MTSL) is used in (Sun et al., 2015) to represent safety requirements of a train control system. Computation Tree Logic (CTL) is a branching-time logic which is employed by five studies. The timed extension of CTL called Timed Computation Tree Logic (TCTL) uses a clock variable to reason about system behaviours over time. TCTL is used to specify the timing and functional requirements of unmanned aerial vehicles in (Misson et al., 2019), which were verified by the UPAAL model checker.

B Language and Z notation are two formal specification model-based languages based on set theory. These can be compared in terms of object orientation, concurrency, tool support and their industrial applications in (Kaur & Gulati, 2012). (Mashkoor & Hasan, 2012; L. Zhang, 2011) apply Z notation as a formal specification language. In (Mashkoor & Hasan, 2012) Object-Z, which extends Z notation with object-oriented features, is used. Event-B, a derivation of B language, has been used in the study (N. K. Singh & Wang, 2019). AADL is used as a specification language in (Hissam et al., 2015; E. Ahmad et al., 2015). In (L. Zhang, 2013a), an extension of AADL called AO4AADL is defined for the specification of aspect-oriented systems. Similarly, Multirate Synchronous AADL language is used in (Bae et al., 2015) for multirate synchronous systems. Another version of AADL is EAST-AADL, which is used for automotive embedded systems. It has been used in (Kang, Mu et al., 2018) whereas its probabilistic extension is utilized in (Kang, Huang & Mu, 2018). SMV is a state-based formal languages and is used in (Drozdov et al., 2017) to represent IEC 61499 model of a distributed automated cyber-physical system.

ICPS typically feature high concurrency. Communicating Sequential Processes (CSP) is a formal language based on calculus that allows modelling concurrency (Roscoe, 1998). CSP has found its application in two primary studies (L. Zhang, 2013c, 2013a) in the form of Timed CSP (TCSP) to specify timing properties for ICPS in transportation.

Apart from traditional formal specification languages, some Domain-Specific Languages (DSLs) have been introduced over the years to accommodate domain-specific requirements formally. Although DSLs have limitations in applicability the solutions they create for a particular domain can be re-used in that specific domain. We find that four primary studies use DSL. A value specification language, variant of MARTE, is used to analyse the requirements engineering process of

ICPS by using stereotypes and annotations in (Ribeiro et al., 2016) to model industrial packing ICPS. A language called ASLan++ Language, an extension of Aslan Formal Specification Language, is used in (Rocchetto & Tippenhauer, 2017) which consists of assertions to identify different classes of attacks on water treatment system. Restricted Test Case Modeling (RTCM) language can be used to generate test cases based on formalised rules (Yue et al., 2015). In (Bouskela et al., 2017), a new language FORM-L is introduced to describe temporal constraints that closely relates to LTL.

Requirements V&V:

A. Languages: Table 3.10 shows the languages used in V&V. These include Hybrid CSP (HCSP), a variant of CSP, which is used in (Zhan et al., 2019) for verifying an Isolette system. Simulink models are translate into HCSP to carry of verification of a train control system in (E. Ahmad et al., 2015). Real-Time Maude

Table 3.10: Languages, techniques and frameworks in requirements verification and validation

Methods	**Types**	**Primary Studies**
Languages	HCSP	(Zhan et al., 2019), (E. Ahmad et al., 2015)
	Real-Time Maude	(Bae et al., 2015), (Meseguer & Ölveczky, 2012)
	LTL-STL	(Nejati et al., 2019), (Nuzzo, Li, Sangiovanni-Vincentelli, Xi & Li, 2019), (Ezio, N, L, Cristinel & D, 2019)
	LTL-MTL	(N. Li et al., 2020)
	DSL-ETL	(Bouskela & Jardin, 2018)
Techniques	Formal Contract	(Nuzzo et al., 2019)
	CIPNs	(Wisniewski et al., 2020)
	State invariants	(Adepu & Mathur, 2016c)
	Cause-effect graphing	(Kim, Chon & Park, 2019)
	Knowledge based (quantitative fitness functions)	(Nejati et al., 2019)
	CPSDebug	(Ezio et al., 2019)
Frameworks	SOCRaTes	(Menghi et al., 2019)
	Unified Graphical Framework	(Zhan et al., 2019)
	Formal Framework	(Dang, Mady, Boubekeur, Kumar & Moulin, 2016)
	Assume-Guarantee Contract Framework	(Nuzzo et al., 2019)

(Ölveczky & Meseguer, 2007) is a language and tool for the formal specification

and verification of real-time systems and is a dialect of Maude (Clavel et al., 2007). It is used in (Bae et al., 2015; Meseguer & Ölveczky, 2012) to verify safety and timing requirements of avionics systems. Metric Temporal Language (MTL), a variant of LTL, is used in (N. Li et al., 2020) to formally validate timing and performance requirements of emergency response missions in the smart cities. Likewise, a domain-specific language called Extended Temporal Language (ETL) is used in (Bouskela & Jardin, 2018) to formally verify the timing requirements of traffic lights and used Modelica, as well, to simulate these requirements.

B. Techniques: Techniques like formal contracts in (Nuzzo et al., 2019), Control Interpreted Petri Nets (CIPNs) in (Wisniewski et al., 2020), CPSDebug in (Ezio et al., 2019), state invariants in (Adepu & Mathur, 2016c) and a graph-based technique called cause-effect graphing is used in (Kim et al., 2019) for verification purposes, as shown in Table 3.10. A Petri Net is a form of bipartite graph and a formal mathematical modelling tool. Its graphical representation enables the visualization of state changes in a system during the runtime because of and they are used to model event-driven distributed computer systems (J. Wang, 2007). In (Wisniewski et al., 2020), CIPNs (variants of Petri nets) are used to determine the safety aspects of a beverage production and distribution system while its functional requirements are specified in CTL. A model checker then verifies these specifications. Model checking and theorem proving (Kallel, 2011) are formal verification methods which are implemented in model checkers and theorem provers. Furthermore, model checking is a technique in which a state-machine model of a system can be automatically analysed by an algorithm to verify whether the model satisfies requirements stated as temporal logic properties. In theorem proving, the correctness of system is proved by building mathematical proofs, often in a semi-automatic manner.

Requirements validation of ICPS requirements is performed by testing, as shown

in Table 3.10. Test models in (Nejati et al., 2019) use Signal Temporal Logic (STL) and convert it into knowledge-based quantitative functions to determine failures in models based on a large number of sampled test inputs. In the same way, the CPSDebug technique presented in (Ezio et al., 2019) is used for testing the functional and fault tolerance requirements of ICPS specified in STL via CPSDebug testing tool (Bartocci et al., 2020).

C. Frameworks: Frameworks like Simulink Oracles for CPS Requirements with uncertainty (SOCRaTeS) (Menghi et al., 2019), Unified Graphical Framework (Zhan et al., 2019), Formal framework (Dang et al., 2016) and an Assume-Guarantee Contract Framework (Nuzzo et al., 2019) have been used for requirements V&V. In (Menghi et al., 2019), Signal First Order logic (SFOL) is used to specify requirements and then test oracles, specified in Simulink, are used to test the ICPS. Formal framework addresses requirements testing along with simulation in (Dang et al., 2016). In this framework, simulation traces are described in the form of a tree to map input signals to the output signals in an industrial HVAC system. The Assume-Guarantee Contract Framework presented in (Nuzzo et al., 2019) uses Stochastic Signal Temporal Logic (StSTL), a variant of STL language, to formalise the specifications and verify the functional and probabilistic requirements of an aircraft electric power distribution system by using the SCANS simulation tool.

D. Tools: Several tools have been utilized for performing V&V, as shown in Table 3.11. Twenty-three primary studies used model checking and seven were found to employ theorem provers.

In addition to traditional formal verification methods, various requirements validation and domain-specific tools have been used in primary studies, as shown in Table 3.12.

For requirements validation purpose, tools such as MaTeLo in (Seceleanu et al.,

Table 3.11: Models checkers and theorem provers employed in primary studies

Tools	Name	Primary Studies
Model Checkers	SMV/nuSMV	(Drozdov et al., 2019),(Gawanmeh et al., 2017), (Bernardi et al., 2020)
	Zot	(Askarpour et al., 2019)
	QVTrace Tool	(Nejati et al., 2019)
	UPAAL	(Kumar et al., 2012),(Balasubramaniyan et al., 2016), (Seceleanu et al., 2017), (Kang, Huang & Mu, 2018), (Kang, Mu et al., 2018),(Kim et al., 2019), (Misson et al., 2019), (L. Huang et al., 2019)
	UPAAL-Statistical Model Checker	(Mancini et al., 2018), (Clarke & Zuliani, 2011), (Kang, Mu et al., 2018), (D. Du et al., 2018), (N. Li et al., 2020)
	nuXmv	(Drozdov et al., 2017), (Grobelna, 2020), (Wisniewski et al., 2020)
	Alloy Analyzer	(Adepu et al., 2016)
	CoPS	(Akella & McMillin, 2009)
	MR-SynchAADL	(Bae et al., 2015)
	Spin	(Drozdov et al., 2019)
Theorem Prover	HOL4	(Mashkoor & Hasan, 2012), (Sanwal & Hasan, 2013)
	HOL Light	(Rashid & Hasan, 2020)
	TVEC Theroem Prover	(Denno & Blackburn, 2014)
	Hybrid Hoare Logic (HHL) Prover	(Nägele et al., 2019)
	KeYmaera X	(Garcia, Mitsch & Platzer, 2019),(Knüppel et al., 2020)

Table 3.12: Requirements validation and domain-specific tools used in primary studies

V&V Tools	Tools Name	Primary Studies
Requirements Testing Tools	MaTeLo	(Seceleanu et al., 2017)
	Test Oracle	(Menghi et al., 2019)
	Toucan4Test	(Yue et al., 2015)
	CPSDebug	(Bartocci et al., 2020)
Domain-Specific Tools	CHASE	(Nuzzo et al., 2018)
	CL-Atse	(Rocchetto & Tippenhauer, 2017)
	HyPLC	(Garcia et al., 2019)
	xSAP	(Ferrante, Di Guglielmo, Senni & Ferrari, 2017)

2017), test oracle in (Menghi et al., 2019), CPSDebug in (Bartocci et al., 2020) and Toucan4Test in (Yue et al., 2015) are used. In four primary studies, (Nuzzo et al., 2018; Rocchetto & Tippenhauer, 2017; Garcia et al., 2019; Ferrante et al., 2017), researchers implement domain-specific tools such as CHASE, CL-AtSe, HyPLC and xSAP for verification. Furthermore, different simulation and co-simulation tools, depicted in Table 3.13, are used by 12 primary studies.

Requirements Management: Requirements are managed not only during RE but also throughout the system development life cycle. Key requirements

Table 3.13: Simulation and co-simulation tools employed during requirements verification and validation

Tool	Name	Primary Studies
Simulation	Simulink	(L. Zhang et al., 2013), (Nägele et al., 2019), (N. K. Singh & Wang, 2019), (Kim et al., 2019), (Clarke & Zuliani, 2011), (Menghi et al., 2019)
	Simulink Design Verifier	(Lin et al., 2018), (Kang, Huang & Mu, 2018), (Kang, Mu et al., 2018)
	SCANS	(Nuzzo et al., 2019)
	Modelica	(Bouskela & Jardin, 2018)
	3D Siemens's Solid Edge ST9	(Metsälä et al., 2017)
Co-Simulation	Modelica	(Clarke & Zuliani, 2011)
	CIROS	(Metsälä et al., 2017)

management concerns are maintaining traceability between requirements and other requirements or system components and handling changes in requirements during system evolution. Table 3.14 shows that only nine primary studies focus on this key activity.

Table 3.14: Techniques and framework in requirements management activity

Type	Methods	Type	Primary Studies
Techniques	Requirements Traceability	Traceability Matrix	(Denno & Blackburn, 2014)
		Contracts graph	(Westman & Nyberg, 2014)
		Petri Net Process	(J. Huang, Zhu, Cheng, Lin & Chen, 2016)
		Traceability Graph	(Sinha, Dowdeswell, Zhabelova & Vyatkin, 2018)
	Non-Conflicting Checks		(Goorden et al., 2019)
	Change Management		(Kulvatunyou et al., 2014), (B. Lima & Faria, 2018)
Frameworks		TORUS	(Sinha et al., 2018)
		Formal Requirement Model	(Jue, Yineng, Wu & Dai, 2019)
		Model Analyzer Framework	(Michael, Atif, Andreas & Alexander, 2020)

A. *Techniques*: Requirements traceability is carried out in (Denno & Blackburn, 2014; Westman & Nyberg, 2014; J. Wang, 2007; Sinha et al., 2018) by different methods such as traceability matrix, contract graphs, process Petri Nets and traceability graphs. A traceability matrix is used in (Denno & Blackburn, 2014)

to generate the test vectors for requirements-to-test traceability automatically. The safety requirements of a fuel level display system are traced through contract structure, a graph-based approach, in (Westman & Nyberg, 2014). A generalized form of Petri Net, called process Petri Net, is used to provide traceability solution for manufacturing activities of bee products (J. Huang et al., 2016). Non-conflicting checks are employed in (Goorden et al., 2019) to detect the impact of requirements splitting on modules.

The process of change management (categorised as a technique in this study) is adopted by (Kulvatunyou et al., 2014) and (B. Lima & Faria, 2018). The change in high-level functional requirements of a reference functional ontology is reconfigured automatically in (Kulvatunyou et al., 2014). In (B. Lima & Faria, 2018), a triage-based automatic personnel allocation system in hospital waiting rooms is implemented.

B. Frameworks: In (Sinha et al., 2018), a framework called Traceability of Requirements Using Splices (TORUS) is presented for the development of large-scale safety-critical CPS, which is based on a traceability graph (a graph-based structure) to create and manage the trace links between requirements and components of smart grids. Similarly, JavaScript Object Notation (JSON) is used to establish a formal requirement model in (Jue et al., 2019) to modify and trace the requirements. JSON is a lightweight, text-based, language-independent data interchange format (Bray, 2017) that is integrated with semi-formal natural language input format. Similarly, in (Michael et al., 2020), a Model/Analyzer Framework is developed for traceability, consistency checking and impact analysis of safety requirements during the development of ICPS.

Classification of ICPS software requirements

Software requirements are derived from system requirements and can be categorised as *functional* and *quality* (non-functional) requirements, according to The Guide to Software Engineering Body of Knowledge (SWEBOK) (Bourque & Fairley, 2014). Functional requirements describe features (the “what") while quality requirements deal with the qualitative aspects of how functional requirements are achieved (the “how well"). We report the results of categorising requirement types in the surveyed works by using the requirements classification described in SWEBOK, as shown in Fig. 3.5.

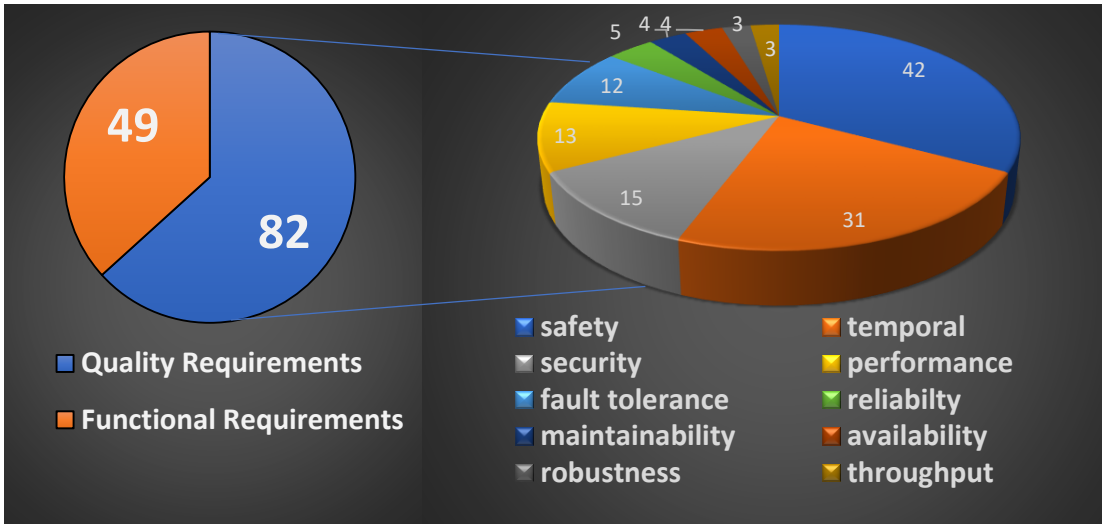


Figure 3.5: Functional and quality requirements targeted in primary studies

Fig. 3.5 shows that 82 (88%) of the studies emphasise quality requirements while around 49 (52%) papers focus on functional requirements. Some studies, like (Vogel-Heuser et al., 2014), (Kang, Mu et al., 2018) and (N. K. Singh & Wang, 2019), address both functional and quality requirements in a single study. Among those targeting quality requirements, safety was considered in 42 (45%) papers followed by timing requirements (31, 33%). This shows that safety and timing are both critical and also interdependent in ICPS. Comparatively, least number (3,

3%) of the studies discuss robustness and throughput requirements.

3.5.2 Findings on the Applicability Of Primary Studies

This section focuses on the evidence of the credibility of the surveyed works and answers the following research questions:

- RQ2.1 Which application domains are used to determine the applicability of selected primary studies?
- RQ2.2 Which research methods and research types have been employed in the selected primary studies?

Application domains

Application domains of the 93 primary studies were analysed separately. As shown in Fig. 3.6, wind-turbine, oil and gas, health care, water treatment, smart grid (Saleh, Althaibani, Esa, Mhandi & Mohamed, 2015), smart city (Gracia & García, 2018), smart manufacturing (Monostori et al., 2016), avionics & aerospace and transportation are the nine main application domains.

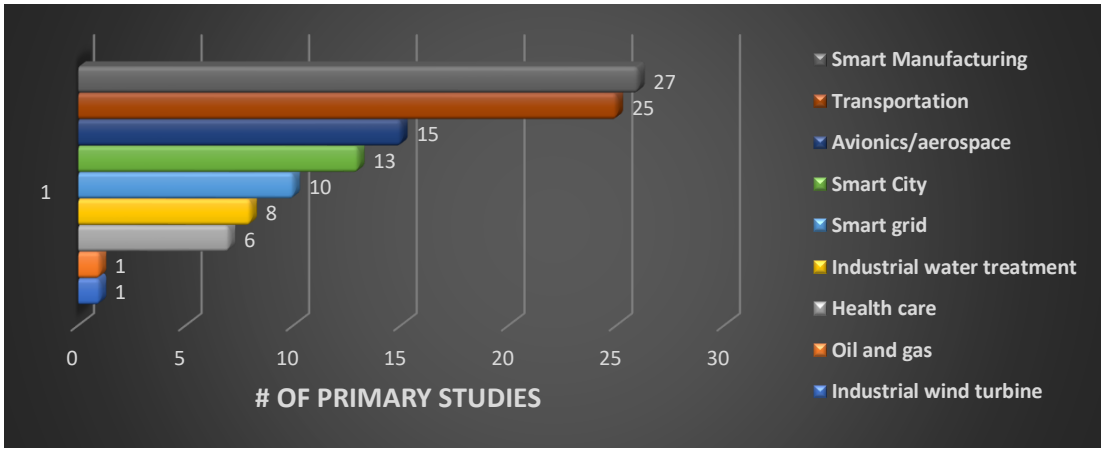


Figure 3.6: Categorisation of primary studies based on application domains

27 (29%) studies (B. Xu & Zhang, 2013; Michael et al., 2020; L. Huang et al., 2019; Grobelna, 2020; Wisniewski et al., 2020; Denno & Blackburn, 2014; Gräßler et al., 2020; Dang et al., 2016; Goorden et al., 2019; Drozdov et al., 2019; Askarpour et al., 2019; Neghina et al., 2019; Y. Chen et al., 2018; Balasubramaniyan et al., 2016; Gawanmeh et al., 2017; Drozdov et al., 2017; Rashid & Hasan, 2020; Metsälä et al., 2017; Iglesias et al., 2017; Ribeiro et al., 2016; Akkaya et al., 2016; F. Li et al., 2016; R. Wang et al., 2011; J. Huang et al., 2016; Zhan et al., 2019; von Birgelen & Niggemann, 2018; Vogel-Heuser et al., 2014) target the domain of smart manufacturing which was the most prominent ICPS domain. It was followed by transportation systems covered in (25, 26%) studies (Jue et al., 2019; Mühlfelder, 2018; Loucopoulos et al., 2019; Kang, Huang & Mu, 2018; Dang et al., 2016; Vogel-Heuser et al., 2014; L. Zhang et al., 2013; Ye-Jing et al., 2013; L. Huang et al., 2019; Mashkoor & Hasan, 2012; Kumar et al., 2012; Clarke & Zuliani, 2011; Bu et al., 2011; L. Zhang, 2011; Akella & McMillin, 2009; Westman & Nyberg, 2014; L. Zhang, 2014, 2013b, 2013c, 2013a, 2013d; B. Xu & Zhang, 2013; Goorden et al., 2019; Knüppel et al., 2020; Bernardi et al., 2020). The lowest consideration was given to the oil and gas sector (Yue et al., 2015) and to industrial wind-turbines with just 1 (1.2%) study (Seceleanu et al., 2017) targeting each domain.

Research methods and research types

Fig. 3.7 classifies the primary studies by research types (Wieringa, Maiden, Mead & Rolland, 2006).

We divide the studies into six categories: solution proposal, evaluation research, validation research, philosophical papers, opinion papers and personal experience paper.

According to the figure, some papers span more than one category. For example,

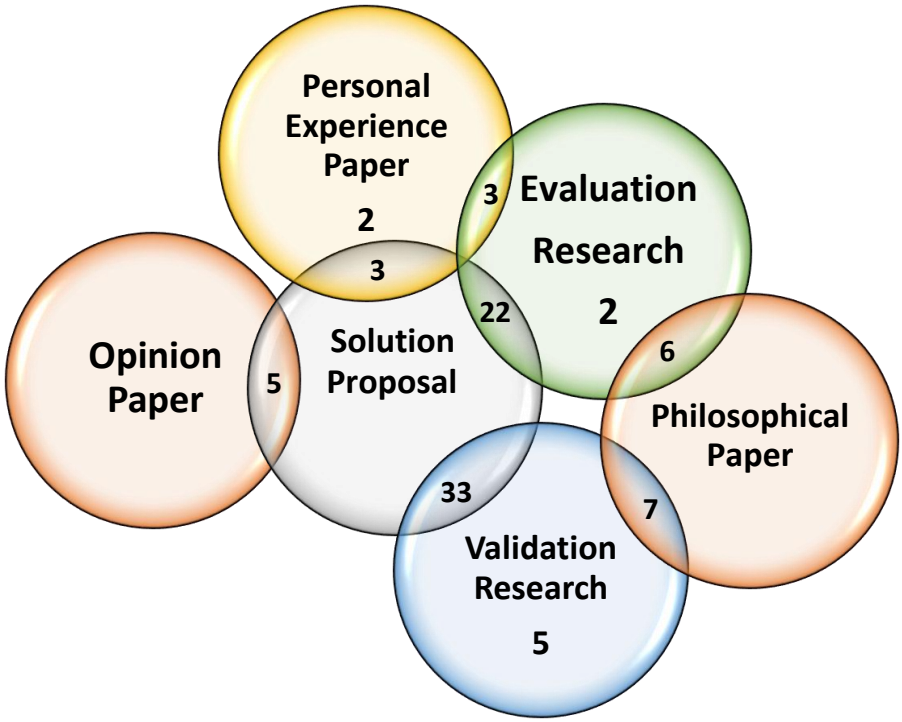


Figure 3.7: Categorisation of primary studies based on research types

primary studies (Sanwal & Hasan, 2013) and (Kulvatunyou et al., 2014) appear in the solution proposal, validation research and opinion paper categories. The most prominent distinction worth mentioning is between validation and evaluation research (Wieringa et al., 2006). Papers classified as evaluation research include case studies, experiments with practitioners, action research, and variations of these methods. In contrast, validation research proposes a novel solution such as mathematical analysis, proof of concept, simulation and prototyping. Solution proposal papers contain novel ideas or techniques that lack full validation or provide evidence on the noteworthy enhancement of existing techniques. Papers with new frameworks are classified as philosophical papers and opinion papers typically report information from leading experts. In personal experience papers, authors report on their own experiences. 63 out of the 93 primary studies were characterized as solution proposals. 33 of these present proof of validity and

therefore fall in both solution and validation research categories. 13 papers lie in the group of philosophical papers. Out of these 7 studies contain validation and the other 6 feature empirical validation. Evaluation research converges with the solution proposal and personal experience classes in 22 and 3 primary studies, respectively. The (5) opinion papers also present solutions and hence overlap with the respective category.

Besides the research type, Fig. 3.8 shows that the use of different *research methods* and experimental setups (Wieringa et al., 2006) in the primary studies helped us in deciding the methods of the research. Overall, case study is the

Figure 3.8: Research methods in primary studies

dominant research method, whereas, experiments and prototyping are the least considered methodologies. Fifty-five (59%) primary studies used case study as a research method. On the other hand, simulation and mathematical analysis or proof of concept are carried out in 20 (21%) and 17 (18%) different studies, respectively.

3.5.3 Bibliography Mapping

This section reports findings corresponding to the following research questions.

- RQ3.1. Which industrial standards have been adopted in identified primary studies?
- RQ3.2 How can the identified primary studies be classified according to the publication years?
- RQ3.3 What are the publication sources and venue types for the identified primary studies?

Classification of primary studies w.r.t industrial standards

The industrial standards adopted in primary studies, shown in Fig. 3.9, have been divided into modelling standards (Sanford, Dov & Yaniv, 2020), production system engineering standards (Y. Lu, Morris & Frechette, 2016) and regulatory standards. 35 (38%) of the primary studies claim to use different modelling standards followed by 11 (12%) that use production system engineering standards. Technologies integrated for manufacturing industries have to comply with regulatory standards. However, only 4 (4%) of the studies cover regulatory standards (Westman & Nyberg, 2014; Sinha et al., 2018; Gomez et al., 2020; Bernardi et al., 2020).

Figure 3.9: Adopted industrial standards in primary studies

Classification of primary studies by publication years

The line graph in Fig. 3.10 shows the number of publications each year. Overall, there has been a moderate increase in numbers over the last 10 years. 2013 saw the first spike of activity with the publication of 9 studies. In 2019, the maximum number of relevant studies (19, 20%) were published.

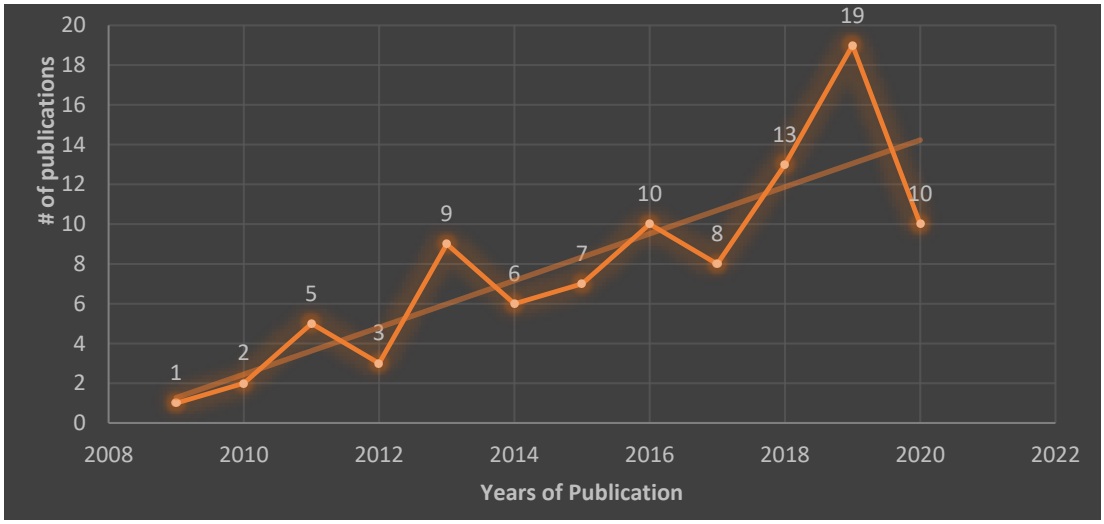


Figure 3.10: Number of primary studies published per year

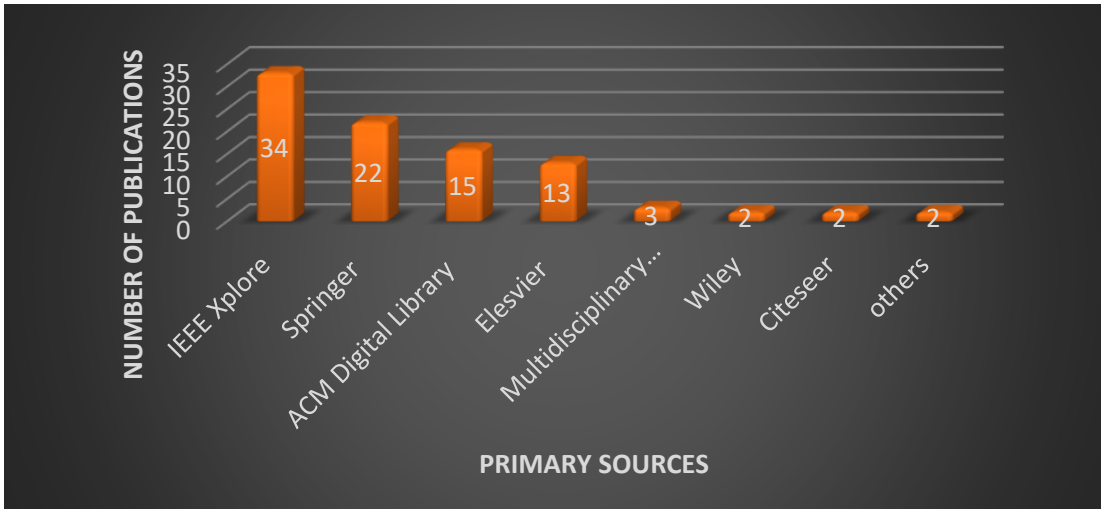


Figure 3.11: Number of primary studies published by publication sources

Classification of primary studies by publication sources and venue types

Fig. 3.11 shows the breakdown of the selected primary studies according to published sources. 34 (34%) of the publications were published in IEEE Xplore, followed by Springer, who published 22 (28%) of the relevant articles. Fig. 3.12

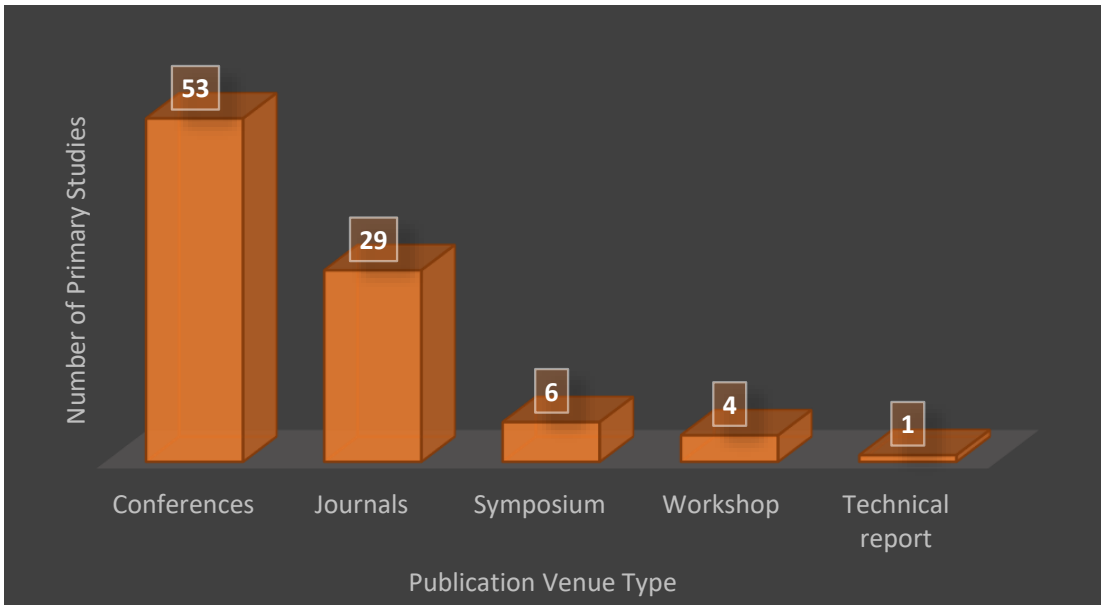


Figure 3.12: Number of primary studies published by publication venue type

shows that most works 53 (57%) were published in conferences, followed by journals. A low number (1%) of relevant technical reports or whitepapers is another indication of low industry adoption.

3.6 Discussion

3.6.1 A Quality Assessment Of This Study

We evaluate the quality of our systematic mapping study by the quality criteria measures QC1–5 presented in (Khan, Sherin, Iqbal & Zahid, 2019). Each measure is given a score of 0, 0.5 or 1 for a SMS. The total quality score for a study is the

sum of its individual scores, which is quantified as low ($0.5 \leq$ quality score ≤ 2), medium ($2.5 \leq$ quality score ≤ 3), or high ($3.5 \leq$ quality score ≤ 5).

The scoring for our SMS based on these criteria is computed as follows.

1. QC1. *Inclusion and Exclusion criteria* have been clearly defined, so we score this measure at 1.

2. QC2. *Search adequacy* is demonstrated by using four reputed digital libraries, resulting in a score of 1.

3. QC3. An explicit *synthesis method* based on a well-used methodology has been presented, resulting in a score of 1 for this measure.

4. QC4. *Quality assessment of included primary studies* was conducted but not reported, so we score this measure at 0.5.

5. QC5. *Information about the primary studies* is provided for each primary study, resulting in scoring this measure at 1.

The overall score of 4.5 means that this SMS lies in the high-quality category. Generally, as the purpose of mapping is to give a broad overview of research, so this quality assessment does not rely on a qualitative assessment of the selected primary studies (Kitchenham et al., 2010).

3.6.2 Gap Analysis

Our survey of 93 primary studies selected out of 3,645 research papers using a multi-phase methodology (Section 3.4) leads to some interesting observations that reveal both current trends and areas requiring further research.

Observation 1: *Most studies explore formal methods with an increasing number looking at the integration of formal and semi-formal methods.*

Meta-analysis: Only a few (eleven) focus solely on semi-formal methods, while 31 studies integrate formal and semi-formal methods. SFM are easier to use and can reduce the time required for producing requirement specifications. FM are more structured and can be automated, but require user expertise and may result in high costs for specifying requirements. The integration allows for a balanced approach where the level of formality can be chosen depending on the criticality of the requirements being handled (Hall, 2005).

Observation 2: *Most studies focus on model-based techniques using formal analysis or verification methods, simulation/co-simulation, agents, model checking and model testing, domain-specific models, semi-formal methods and iterative approaches.*

Meta-analysis: The surveyed works use model-based techniques for different RE activities. This shows that while model-oriented paradigms for ICPS are growing, but there is no accepted standard for modelling and evaluating ICPS (Derigent, Cardin & Trentesaux, 2020). Owing to the closed-coupling of hardware, software, and physical structure of ICPS, model-based techniques have to consider three perspectives uniformly: functionality (implemented in software), physicality (physical environment and hardware platform), and architecture. Nevertheless, most of the current model-based techniques do not cover all of these three aspects uniformly.

Observation 3: *Most works focus primarily on safety and timing requirements.*

Meta-analysis: The dynamic nature of ICPS raises challenges regarding the quality requirements of systems which needs consideration at the requirements

stage. Fig. 3.5 shows that more attention should be given to robustness, throughput, maintainability availability and fault-tolerance. Additionally, predictability and self-awareness requirements are not covered at all.

Observation 4: *Studies that use SysML, Modelica and IEC 61499 propose methodologies useful to improve production system engineering. Few works like (Iglesias et al., 2017) aim to reduce manufacturing costs by improving information exchange among suppliers and manufacturers by using ISA 95 Standard. Some works, for example, (Metsälä et al., 2017; LeMay et al., 2011) have tried to develop control and production planning strategies in order to improve robustness, reconfigurability, flexibility and security of ICPS using OPC-UA, system-level iterative approaches, formal analysis techniques or combined formal and semi-formal methods.*

Meta-analysis: Existing design and development standards are far from being sufficient for the ICPS ecosystem because they cannot keep pace with rapidly evolving requirements. New or improved standards are required which have an impact on product and production life cycles, business cycles, and supply chain management in order to improve quality, economy and productivity.

Observation 5: *Most works address multiple, but not all, RE activities.*

Meta-analysis: There is still no standard or generally accepted RE process defined for ICPS.

Observation 6: *Most works focus on only problem or solution domain requirements.*

Meta-analysis: More work is needed to study the interaction between problem and solution domain requirements.

Observation 7: *Formal ontology or domain-specific languages, frameworks and tools help to capture cross-domain relationship, give domain-specific views of requirements at different levels of abstraction and promote the reusability of requirements specification in a particular domain.*

Meta-analysis: Muti-domain integration and migration is a significant challenge in ICPS. More work is needed to provide clear semantics (interfaces) to establish and maintain the relationships between different domains.

Observation 8: *Formal contracts in primary studies are either vertical contracts (design exploration, early detection of errors), horizontal contracts where rules are formalised for subsystems interaction with its environment or stochastic contracts.*

Meta-analysis: Formal contracts for different domains need continuous-time contracts (Fisher et al., 2014) that can not only differentiate between the discrete event and continuous changes but also, based on discrete constraints, can express the bounds on continuous behaviours. Formal contracts for probabilistic requirements are still in the early stages. We also require tools to formalise requirements through contracts effectively and to explore techniques to improve their suitability and scalability.

Observation 9: *Requirements elicitation is a critical RE activity in ICPS but only a few works look at eliciting requirements, removing ambiguities, and elaboration.*

Meta-analysis: Systematic, formal or semi-formal approaches for requirements elicitation and management are urgently needed in large-scale distributed ICPS. These approaches would need to strike the right balance between expressiveness and rigour to be usable in industry (Jeon, Yoon, Um & Suh, 2020).

Observation 10: *Privacy and trustworthiness are important quality requirements that are becoming increasingly important in ICPS (Fink, Edgar, Rice, MacDonald & Crawford, 2017). Unfortunately, none of the surveyed works focuses on these requirements.*

Meta-analysis: New formal or semi-formal methodologies that extend legacy methods for these concerns are needed urgently.

Observation 11: *Three papers classified in requirements analysis and one in requirements V&V use co-simulation to analyse and verify large-scale behaviours in the early stages of system development.*

Meta-analysis: FM and SFM in RE complement *simulation* as well as *co-simulation*. Simulation is a well-understood strategy to explore and test systems and finds several uses in the RE of ICPS. (Akkaya et al., 2016) and (Menghi et al., 2019) use simulation to analyse requirements. (Kim et al., 2019) use Simulink Design Verifier (SDV) with the nuSMV model checkers for requirements V&V using simulation. (Kang, Huang & Mu, 2018; Dang et al., 2016) use simulation for both requirements analysis and V&V purpose. Co-simulation and SysML are integrated in (Neghina et al., 2019) which shows that co-simulation can be used with semi-formal foundations, too. Testing for validation purpose involves the use of simulation for requirements testing, like in (Kim et al., 2019) and (Menghi et al., 2019). Co-simulation is a promising *modular* approach to manage complexity in ICPS (Wiesner et al., 2015) and is a priority area for future development.

Observation 12: *Only two studies integrate requirements validation and verification.*

Meta-analysis: The interplay of validation and verification is largely unexplored, with most techniques focusing on only one aspect. Future exploration may reveal

optimisations and efficiencies in taking a holistic and integrated approach towards these closely related aspects of RE in ICPS.

Observation 13: *Most works employ model checking techniques for verification.*

Meta-analysis: Model checking is attractive as it is fully automated. However, user-guided verification like in theorem proving may be more desirable for large-scale ICPS where model checking does not scale. Model checking and proof-theoretic approaches can be combined to verify complex requirements of ICPS.

Observation 14: *The empirical evidence of the effectiveness of emerging semi-formal and formal methods is largely missing. The limited number of personal experience papers using formal or semi-formal methods in the industry is also deficient.*

Meta-analysis: Industry adoption is low. Processes for the rapid maturation of lab-based solutions and testing them in industrial settings are needed.

Observation 15: *Interesting combinations of different types of methods have been successfully used for various RE activities. Timed CSP is combined with a cause-effect graph in (Kim et al., 2019), (L. Zhang, 2013b) mix automata with Modelica, and (Metsälä et al., 2017) use abstract state machines illustrated using 3DSiemens' Solid Edge ST9.*

Meta-analysis: These groupings indicate that the availability of specialised methods enables novel integration that may be better suited than individual methods. For instance, Modelica is a popular method to model the physical part of an ICPS and can be combined with AADL or UML to model the cyber aspects, like in (L. Zhang, 2013c).

3.6.3 A Conceptual Model To Aid Practitioners

The primary drivers of the RE process in ICPS are regulatory standards, software requirements (functional and quality requirements), and stakeholders requirements. Furthermore, formalisms in selected studies belong to different illustration styles and programming paradigms. Therefore, RE of ICPS using these formalisms also adopts, indirectly, these illustration styles and programming paradigms. Thus, the relationship between RE of ICPS and formalisms is captured in the form of a conceptual model shown in Fig. 3.13. Adapted from (Lana et al., 2019), this model relates to illustration styles defined in (Mandayam K. & Steven P., 1995) and programming paradigms listed in (Van-Roy & Haridi, 2004). This conceptual model is mapped in Table 3.15 to *RE activities, formalisms* (formal/ semi-formal), *types of methods* (methods can be semi-formal or formal languages or techniques. Frameworks and tools are not included here because they inherit the characteristics of the languages and techniques that they support, such as *illustration style* and *programming paradigm.*

Table 3.17: Advantages and disadvantages of formal and semi-formal techniques

Techniques	**Advantages**	**Disadvantages**
Prototype	allows iterative development of ICPS for better understanding and communication among stakeholders	complexity and scale of ICPS make prototyping harder, impractical to prototype the physical system whose existence is unknown in advance
Abstract State Machine	enables high level analysis and design, effective when different analysis and validation techniques may be applied to the same model	not easy to construct accurate ground models of requirements
Hybrid Automata	describes systems with mixed continuous and discrete dynamics	provides limited support to represent nonlinear and spatial-temporal features of ICPS

continued ...

Table 3.17: Advantages and disadvantages of formal and semi-formal techniques

Techniques	**Advantages**	**Disadvantages**
Cellular Automata	very efficient in model spatial-temporal requirements of ICPS, allows efficient parallel computation	emergent behaviors of ICPS can lead to redundant results, not suitable for the environment that generates unpredictable results
Graph Theory	models the topology of ICPS, demonstrate spatial-temporal requirements	cannot describe heterogeneity of nodes
Formal Contract	enhance reusability of specification, efficient for large and hybrid design-space exploration	has scalability problem in case of probability requirements
Formal Ontology	use to overcome cross-domain barrier, efficient for semantic interoperability	data is structured in such a way that it does not allow to add inconsistent data which can be made consistent later
Cause-effect Graphing	detect the ambiguity and incompleteness by generating testcases.	expertise is required, need to be very focused, difficult to work with large specifications
CIPNs	efficient for communication with environment by signals, concurrency can be shown graphically	need expertise to understand methodology
VDM-RT	model timings constraints of ICPS, support asynchronous analysis	VDM models are not accurate in the sense of physical implementation
Stochastic Timed Automata	generic technique to model spatial behaviours in various domains and provide accurate semantics of requirements	result in random delays, unlikely behaviours can be ignored due to selection of interactions.
PDRTA	model the discrete events of ICPS based on probability	does not perform well in practice
State invariant	validates the states of ICPS components, helps in making a decision and monitoring cyber process	As ICPS have the mass of state invariants so one fault in the system may result in an abundance of broken invariants.
Traceability Graph	useful for multi-level tracing, visualisation tool for a large set of requirements	Modelling effort and ensuring coherence between requirements and graph models
Traceability Matrix	helps the testing team to understand the level of testing activities done for the specific product.	not suitable for higher dimensions trace links
Process Petri-Nets	manage the complexity of ICPS by providing a clear separation of technological model and resource sufficiency	are system dependent, are not equipped with a notion of physical distribution, do not portray self-organizing cyber and physical production

continued . . .

Table 3.17: Advantages and disadvantages of formal and semi-formal techniques

Techniques	**Advantages**	**Disadvantages**
Use Cases	user-oriented, manage the complexity, provide a base to specify end-to-end temporal requirements.	For ICPS having infinite interactions with its environment, a large number of use cases have to be created. If creations are limited to only important scenarios, then few use cases lead to insufficient specification.
Domain Model	gives a better understanding of the complex domain, helps in improving communication among teams	time-consuming, requires domain expertise
Requirement Diagram	helps in analysis and traceability of formal and quality requirements.	lack precise decomposition semantics, relatively immature diagram, relationships allocation are incomplete and ambiguous
Meta-Model	helps in understanding and describing the large system, supports reusability, assure consistency among teams	can be difficult and challenging to define right abstractions and structure them for reusability, face compatibility problems among multi-domains, time-consuming
Parametric Diagrams	models the constraints and mathematical relationship between components in order to fulfil the performance requirements.	Parametric constraints are not clearly understandable, immature as compared to other SysML diagrams.

The findings of this study indicate that the illustration style hierarchy adopted by the primary studies can be divided into two sub-styles: Property-Oriented (PO) or Model-Oriented (MO). PO sub-style depicts the properties of the system at a higher level of abstraction resulting in the less-detailed specification. It can be further categorized into algebraic-oriented and axiom-oriented styles. Axiom-oriented styles include ontology-based, contracts-based, rule-based and knowledge-based programming paradigms. MO is comprised of object-oriented, aspect-oriented, state-based, probability-based, automata-based, language-oriented and discrete-event based programming paradigms.

The findings of this study can be transformed into guidelines for academic and industrial practitioners. For academic practitioners, Section 3.5 determines the

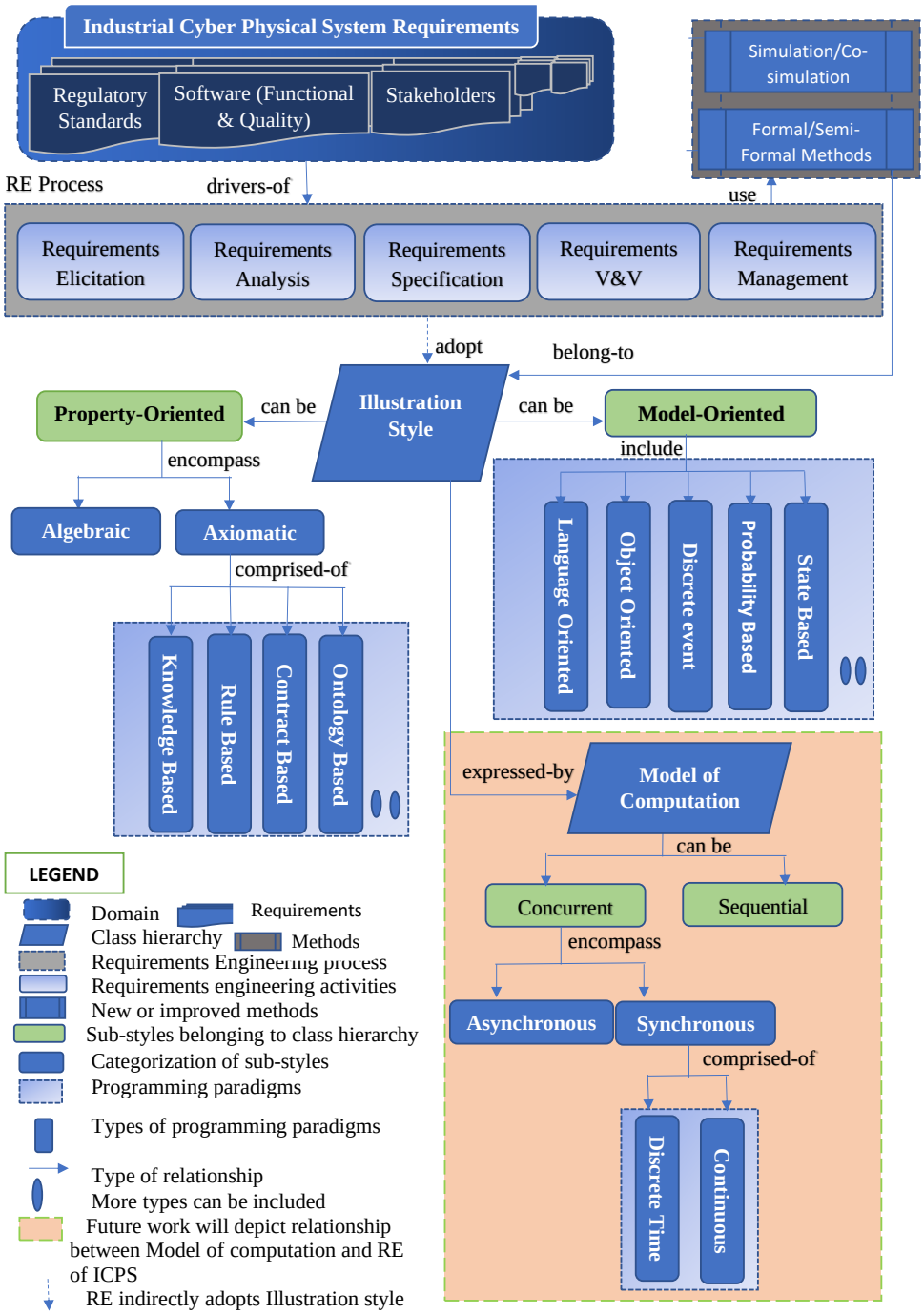


Figure 3.13: Contextualization based on illustration style, programming paradigms and improved methodologies to show the relationship between formalisms and requirements engineering of Industrial cyber-physical system

Table 3.15: Mapping of conceptual model on the basis of requirements engineering activities, formalisms (semi-formal/formal), types of methods (techniques/languages), method name, illustration style (property-oriented/model-oriented), programming paradigms

RE Activities	Formalism	Types of Methods	Method Name	Illustration Style	Programming Paradigm
Requirements Elicitation	Semi-Formal	Techniques	Prototype	MO	Language-Based
			Graphic Notations	MO	Discrete Event
	Formal	Languages	PA	PO	Rule-Based
			Alloy	PO	Rule-Based
	Semi-Formal		AADL	MO	Object-Oriented
			UML		Object-Oriented
			SysML		Object-Oriented
			MARTE		Object-Oriented/ Language-Based
			Modelica		Object-Oriented
			POOSL		Object-Oriented
Requirements Analysis	Formal	Techniques	Abstract State Machine	MO	Automata-Based
			Hybrid Timed Automata		Automata Based/ State Based
			Cellular Automata		Discrete-Event
			Graph theory	PO	Automata-Based
			Formal Contract		Contract-Based
			Formal Ontology		Ontology-Based
			VDM-RT		Discrete-Event
			PDRTA	MO	Automata-Based/ Probability-Based
			Stochastic Timed Automata		Probability-Based
	Semi-Formal		Use-Cases	MO	Object-Oriented
			Domain Model		Object-Oriented
			Meta-Model		Object-Oriented
			Object Diagram		Object-Oriented
			Parametric Diagram		Object-Oriented
Requirements Specification	Formal	Languages	Event B	MO	State-Based
			Z Language		State-Based
			Object Z	MO	Object-Oriented/ Aspect-Oriented
			Temporal Logic	PO	Rule-Based
			CASL		Algebraic
			SMV	PO	State-Based
			First Order Logic		Rule-Based
			ASLan++	MO	Language-Based
			Timed CSP	PO	Rule-Based
			Value Specification Language	MO	Language-Based
			RTCM	MO	Language-Based
			Form-L	PO	Rule-Based/ Language-Oriented
Requirements Verification and Validation	Formal	Languages	Real-Time Maude	PO	Rule-Based
			HCSP	MO	Discrete-Event
			STL	MO	Discrete-Event
			ETL	PO	Rule-Based / Language-Based
		Techniques	Formal Contract	PO	Contract-Based
			State invariants	PO/MO	Rule-Based/ Discrete-Event
			Cause-effect Graphing	MO	Automata-Based
			Knowledge-Based(Quantative Fitness Function)	PO	Knowledge-Based
			CIPNs	MO	State-Based
Requirements Management	Semi-Formal	Technique	Traceability Graph	MO	Automata -Based
			Non-conflicting check	PO	Rule-Based/ Algebraic
			Traceability Matrix	MO	Object-Oriented
	Formal		Process Petri-Nets	MO	Discrete-Event

Table 3.16: Advantages and disadvantages of formal and semi-formal languages

Languages	Advantages	Disadvantages
AADL	model hardware and software components of ICPS, check components consistency in discrete-time	does not model the spatial-temporal features of transportation, does not verify the components in continuous time
UML	help practitioners to tackle complex software structures, model functional structures of software abstracted from inner details of system	not able to customize description rules for extended requirements, modeling and management of physical resources and concurrency are challenging
SysML	general-purpose modelling language used for specification, analysis, design, V&V of ICPS	unable to represent mechanical, physical components and description of system resources through static and dynamic diagrams
MARTE	model hardware and software parts of ICPS, provide interoperability between different tools of used for specification, design and verification	does not have specific methodology, physical models are not modelled by it
Modelica	model and simulate physical parts of ICPS, increase robustness, involve in hybrid and multi-domain modelling	graphical formalisms are not available, hard to find programming and modelling errors
POOSL	expressive in nature, model for analysis of ICPS, supports flexible and reusable designs.	unable to combine inheritance and concurrency in a flexible way
PA	suitable to describe the order of occurrence of events, facilitate the modular composition of process.	unable to handle complex operations, does not support performance and function analysis
Alloy	have strong analytical capability, very expressive in defining complex structure and behavior of ICPS	limited support for numerical constraint, lack built in support for dynamic system modelling
Event-B	used for system level modelling and analysis of ICPS, provides different level of abstraction of system, support formal refinements, can prove timing requirements	stochastic behavior are not supported well
Object Z	express complex data operations, support modularity	operations are atomic, no direct way to determine that how much time an operation will take to complete
LTL	expressive, formal and compact notation to state safety and liveness requirements	cannot represent the non-deterministic state or transitions
CTL	can represent non-deterministic states or transitions	cannot express fairness requirements directly in formalism, does not cater stochastic features
SMV	provide modular and hierarchical description of ICPS, supports reusability	graphical formalism is not available
Signal	powerful language, can capture time and	non-deterministic exponential time is hard
First Order Logic	magnitude continuous behavior of ICPS, able to handle uncertainties due to ICPS-environment interaction	to monitor
ASLan++	flexible, expressive, easy to use and formally specify security requirements of ICPS	does not cover all modes of encryption for security
Timed CSP	provide the facility to analyse run-time behaviors, has strong ability to model process control	time-consuming, modelling is tedious
RTCM	have user friendly template, set of keywords, and rules for writing test cases specifications.	specific to the application so applicability is limited
FORM-L	cater stochastic aspects and deterministic behaviour , deal with spatiotemporal constraints	need the expertise to understand the language.
Real-Time Maude	easy to use and specify safety and timing requirements of ICPS.	no guarantee of complete search and model checking
HCSP	expressive, easy to use, model hybrid behaviour of ICPS	involve the sequential composition of operation which make interruption difficult to handle
ETL	capable of describing continuous real-time physical aspects and stochastic behaviours	dependent on high-level languages in order to model
STL	formalise control-theoretic properties, express timing constraint	online monitoring is not efficient, optimisation problem, cannot intend for frequency-domain analysis

current trends and gaps within the scope of this study, which are also addressed in Section 3.6.2. For industrial industrial practitioners, Table 3.8, Table 3.11, Table 3.12, Table 3.13, Fig. 3.5 and Fig. 3.6 can assist in identifying state-of-the-art methods that can be adopted. To benefit both academic and industrial practitioners, we present a further comparative analysis between formal and semi-formal languages and techniques in Table 3.16 and Table 3.17, respectively. The agility and reflection of system engineering practices in our conceptual model and its mapping can help both academic and industrial practitioners select formal and semi-formal methods depending on programming paradigms, illustration styles or requirement types.

We illustrate the utility of our conceptual model on a real-world case study of a USB stick production line (Neghina et al., 2019). This case study includes subsystems like Human–Machine interface (HMI), Part Tracker, Warehouse, Robotic Arm, Wagons, Test Station to handle the customers' order. The detailed description of each of these subsystems and experimental report is provided in (Neghina et al., 2019). Our model provides an overview for the practitioners to select the desired methods depending on their needs. For example, if the objective is to specify the timings constraints of ICPS along with an asynchronous analysis of all involved subsystems, we can use VDM-RT as it can produce abstract functional model units for all subsystems. These model units can describe both the physical and cyber parts of the USB production system. Furthermore, to link the model units to customer order requirements, we can examine an object-oriented paradigm, such as SysML that can be used later to configure co-simulation. Similarly, requirements can also be defined as rules and analysed by Process Algebra, while Hybrid Automata can be used to describe both the cyber and physical aspects of the production system. According to these decisions, VDM-RT as a formal technique, SysML as a semi-formal language and co-simulation as a

method to perform requirement analysis, are integrated. VDM-RT and SysML belong to the discrete event and object-oriented programming paradigms and adopt a model-oriented illustration style. Similarly, Process algebra is included in a rule-based programming paradigm that has a property-oriented illustration style while Hybrid Automata follow a state-based programming paradigm using a model-oriented style.

3.7 Conclusions And Future work

As the use of ICPS has grown, researchers and practitioners have become more inclined to employ new or improved requirements engineering methods for developing quality ICPS. However, the scale, heterogeneity, and complexity of ICPS, as well as their evolutionary nature and the involvement of a multitude of stakeholders have made RE of ICPS a challenging task. A comprehensive landscape of methods that can identify, analyse, verify or manage ICPS requirements has been missing. This research addresses this by reporting a systematic mapping study on available formal and semi-formal methods for the RE of ICPS. Semi-formal and formal methods promise rigour and structure that are seen as essential ingredients in building robust, repeatable and scalable requirements engineering processes for building ICPS. The findings of the study result in a novel conceptual model that highlights current trends and research gaps in the area.

Our findings have identified several new research directions as future work: Firstly, comprehensive comparisons of each formal/semi-formal technique, language, tool or method utilised in different activities of requirements engineering can be analysed. Next, co-simulation techniques can be optimised to analyse the different types of requirements such as performance, probability, trust and privacy and fault tolerance for providing customer-specific solutions and for resolving

constraints on ICPS. Thirdly, practitioners can combine model checking and proof-theoretic approaches to verify the complex requirements of ICPS. Also, more work is needed to provide clear semantics (interfaces) to establish and maintain the relationships between different domains. Lastly, formal contracts for probabilistic requirements are still in the early stages and new methods are required to formalise such requirements.

For our future work, we will extend our conceptual model to further analyse the relationship between RE of ICPS and models of computation (MoCs). This analysis will not only articulate their benefits to the industrial community but also help them identify and choose the appropriate MoCs in order to comply with industrial standards. Furthermore, we are developing design patterns to enable easy integration of security requirements from standards into ICPS component and system software. The key challenge in this direction is to sufficiently secure a system without sacrificing performance or overwhelming the limited computation powers of ICPS hardware components like PLCs.

Chapter 4

Prelude - Manuscript 2

The following chapter published as a conference paper in *2021 IEEE 18th International Conference on Privacy, Security and Trust (PST)* under the title of *Light-Weight Active Security for Detecting DDoS Attacks in Containerised ICPS* (Zahid, Kuo & Sinha, 2021a). The chapter introduces a novel, light-weight active security approach to actively detect DoS/DDoS attacks through frequency analysis of incoming network traffic (packets). The prototype implementation and evaluation indicate that the proposed light-weight active security solution is suitable for resource-constrained ICPS.

Manuscript 2 encompasses two research objectives (RO2 and RO4). First, to achieve RO2, we delved into a more qualitative evaluation of key ICPS challenges like active security against cyber attacks, specifically DDoS attacks, in resource-constrained ICPS. We have performed Systematic Literature Review (SLR) of existing approaches for providing active security in such systems (Section. 5.2). The survey focused on the generic and ICPS related DoS/DDoS attack detection strategies. The existing SLR has addressed the studies up to April 2021. Because of the page limitations imposed by the conference, we could not include a tabular summary of this SLR. The tabular summary offers a structured and accessible

way to present the findings, facilitates the identification of the research gaps, and is understandable to both researchers and non-expert audiences. Furthermore, to ensure and enhance the validity that our review incorporates the recent and relevant research in the field and is aligned with the current state of knowledge, we have extended our SLR further until August 2023 (Table. 4.1). The survey findings have identified several research gaps. One of the research limitations is the need for more emphasis on the resource-constrained ICPS. Additional gaps include the use of resource-intensive attack detection methods that are not suitable to the resource-constrained ICPS and rely on time-domain analysis. The solution for the identified research gaps is addressed in the research objective (RO4) by proposing the generic framework for actively securing resource-constrained ICPS in the frequency domain.

Table 4.1: Systematic literature review of existing approaches used for Distributed (Denial of Service) attacks detection in Industrial Cyber-Physical Systems

References	**Detection Method**	**Domain Analysis**	**Attack Type**	**Resource Constraint**
(C. Chakraborty, Nagarajan, Devarajan, Ramana & Mohanty, 2023)	Multi-Source Transfer Learning (CMTL)	Time	DDoS	No
(B. B. Gupta, Chui, Arya & Gaurav, 2022)	Statistical Approach	Time	DDoS	No
(Gowripeddi, Sasirekha, Bapat & Das, 2023)	Digital Twin (DT) approach	Time	DDoS	No
(Yadav & Mishra, 2023)	Lyapunov functions, DL	Time	DoS	No
(Sharma et al., 2023)	Bidirectional Long Short Term Memory (LSTM)	Time	DoS	No
(Gyamfi & Jurcut, 2022)	Long Short Term Memory (LSTM)	Time	DoS	No
(F. Liu, Zhang, Ma & Qu, 2022)	Support Vector Machine	Time	DoS	No
(B. Liu, Chen & Hu, 2022)	Mode division	Time	DoS	No
(Diaba, Shafie-khah & Elmusrati, 2022)	Convolutional Neural Network, Gated Recurrent Unit	Time	DoS	No

continued ...

Table 4.1: Systematic literature review of existing approaches used for Distributed (Denial of Service) attacks detection in Industrial Cyber-Physical Systems

References	**Detection Method**	**Domain Analysis**	**Attack Type**	**Resource Constraint**
(Shimeng et al., 2022)	Denoising autoencoder	Time	DoS	No
(W. Wang, Harrou, Bouyeddou, Senouci & Sun, 2022)	ML	Time	DDoS	No
(Ravi, Chaganti & Alazab, 2022)	Kernel-based principal component analysis (KPCA)	Time	DDoS	No
(Sambangi, Gondi & Aljawarneh, 2022)	Gaussian based	Time, Frequency	DDoS	No
(Lan et al., 2022)	E-minBatch GraphSAG model	Time, Frequency	DDoS	No
(Hao, Yang & Yang, 2021)	Statistical, ML	Time	DoS	No
(Rouzbahani, Bahrami & Karimipour, 2021)	Ensemble Deep Neural Network (SEDNN)	Time	DDoS	No
(Althobaiti, Kumar, Gupta, Kumar & Mansour, 2021)	gated recurrent unit (GRU)	Time	DoS	No
(Haghighi, Farivar, Jolfaei & Tadayon, 2020)	AI, Mathematical model	Time	DoS	No
(Mahmoud, Hamdan & Baroudi, 2020)	Random Conditional Probability	Time	DDoS	No
(Kordestani, Chaibakhsh & Saif, 2020)	AI (neural network)	Time	DDoS	No
(J. Liu et al., 2020)	IDS, Mathematical model	Time	DoS	No
(Tahir, Khan & Asad, 2019)	Linear Matrix Inequality	Time	DDoS	No
(Gao, Chai, Zhang & Xia, 2019)	Model Based, Mathematical model	Time	DoS	No
(Schneble & Thamilarasu, 2019)	IDS, ML	Time	DoS	Yes
(Elgendi, Hossain, Jamalipour & Munasinghe, 2019)	AI	Time	DoS	No
(Su & Ye, 2018)	Linear matrix inequality	Time	DoS	No
(Biron, Dey & Pisu, 2018)	Mathematical model	Time	DoS	No
(Taormina, Galelli, Tippenhauer, Salomons & Ostfeld, 2017)	Attack Model	Time	DoS	No
(A.-Y. Lu & Yang, 2017)	linear matrix inequality	Time	DoS	No
(Ashok, Govindarasu & Wang, 2017)	Model Based	Time	DoS	No
(S. Huang, Zhou, Yang & Qin, 2015)	ML	Time	DDoS	No

continued ...

Table 4.1: Systematic literature review of existing approaches used for Distributed (Denial of Service) attacks detection in Industrial Cyber-Physical Systems

References	**Detection Method**	**Domain Analysis**	**Attack Type**	**Resource Constraint**
(Nizam, Chaki, Al Mamun, Kaiser et al., 2016)	Fuzzy logic	Time	DoS	No
(Vuong, Loukas, Gan & Bezemskij, 2015)	IDS, decision tree, ML	Time	DDoS	Yes
(Krotofil, Cárdenas, Manning & Larsen, 2014)	Model Based, Mathematical model	Time	DoS	No

Chapter 5

Light-Weight Active Security for Detecting DDoS Attacks in Containerised ICPS (Manuscript 2)

5.1 Abstract

In Industrial Cyber-Physical Systems (ICPS), containerisation promises high scalability, reconfigurability and dependability. Denial of Service (DoD/DDoS) is a significant security threat in containerised ICPS applications, which execute on resource-constrained computers like PLCs, and cannot support traditional security mechanisms like firewalls that sacrifice performance and throughput.

We propose a novel, light-weight active security approach to detecting DoS/DDoS attacks through frequency analysis of network traffic (packets). Our approach identifies attacks by recording a frequency signature of the flow of packets in an ICPS under normal operation. Subsequently, an attack is modelled as any

anomalies in the network that modify the frequency profile of network traffic in the ICPS. Our prototype implementation and evaluation show that this active security method is light-weight and suitable for resource-constrained ICPS platforms.

5.2 Introduction

ICPS are driving the 4th Industrial Revolution with a significant impact on all sectors, including industrial automation (Colombo et al., 2014). As system sizes grow and the pace of development continues to increase, ICPS require increasing flexibility in deploying applications for scalability, reconfigurability and dependability. *Containerisation* technologies like Docker (Anderson, 2015) provide well-defined interfaces for multi-domain integration and migration(Zahid, Tanveer et al., 2021), higher operational efficiency and resource optimisation; hence, containerised technology can offer unparalleled flexibility in deploying ICPS applications (Stanciu, 2017).

Communication security is an ever-growing concern in ICPS. Containerised ICPS applications are vulnerable to newer and more sophisticated attacks such as DoS/DDoS attacks that can significantly impact the performance and safety of critical ICPS applications that traditionally execute on resource-constrained computers like PLCs and use low bandwidth networks. Note that there are two categories of DoS attack: high-rate and low-rate (Z. Wu, Yue, Li & Xie, 2015) and low-rate DoS attacks are out of scope of this study.

In the literature, various DoS/DDoS detection schemes have been proposed for ICPS, however, none of the existing works addresses the novel security challenges in containerised ICPS applications. The few techniques that do address the DoS or DDoS attacks in containerised applications(Chelladhurai, Chelliah & Kumar, 2016; Tomar, Jeena, Mishra & Bisht, 2020; Yasrab, 2018) are heavyweight and

therefore not feasible to provide active security in resource-constrained ICPS. Hence, the existing attack detection mechanisms are rigid that either need manual configurations or have limited measures to respond to an attack. There is a need for some detection mechanism that can actively (dynamically and programmability) provide communication security to ICPS applications (Hand, Ton & Keller, 2013). These observations lead to the following research questions:

RQ1 How can light-weight active security mechanisms be provided in resource-constrained ICPS for secure communications?

RQ2 How can the technique proposed in answering RQ1 be evaluated through a prototype for its feasibility?

To the best of our knowledge, this is the first study to provide a light-weight active security approach for detecting DoS/DDoS in resource-constrained containerised ICPS. RQ1 and RQ2 are answered through an adapted Design Science research methodology (Offermann, Levina, Schönherr & Bub, 2009) to build and test a technique to detect DoS/DDoS attacks in resource-constrained containerised ICPS applications. We propose a novel light-weight active security approach to monitor and detect the presence of DoS/DDoS attacks on Docker-based resource-constrained ICPS applications by analysing network traffic (packets) in real-time. We profile the frequency characteristics of traffics by taking advantage of Quick Discrete Fourier Transform (DFT) (H. Guo, Sitton & Burrus, 1994). Frequency domain analysis provides an innate understanding of system behaviour which is often difficult via time-domain analysis. We first establish an expected *frequency signature* as the baseline for a system under normal operation. Network traffic is then monitored in real-time and its DFT output is continually compared to the system's frequency signature. If the system's current frequency pattern deviates sufficiently from the baseline, the system is considered to be under attack.

The primary contributions of this study are: 1) A novel light-weight active security algorithm based on frequency analysis for resource-constrained containerised ICPS (Sec. 5.3). 2) Experimental results (Sec. 5.4), based on a publicly available dataset, that shows the efficacy of the proposed approach.

5.3 Proposed Containerised Attack Detection Model

Our attack detection model is built for resource-constrained containerised applications. We prototype our approach using a system deployed using Docker (server container, multiple client containers and one or more attacker containers), but our overall framework also applies to other containerisation technologies or applications running directly on the devices.

Fig. 5.1 shows the proposed attack detection model that has three major phases: *packet capture*, *pre-processing* and (DoS/DDoS) *attack detection* and where the proposed approach is embedded as an algorithm and executes on the server.

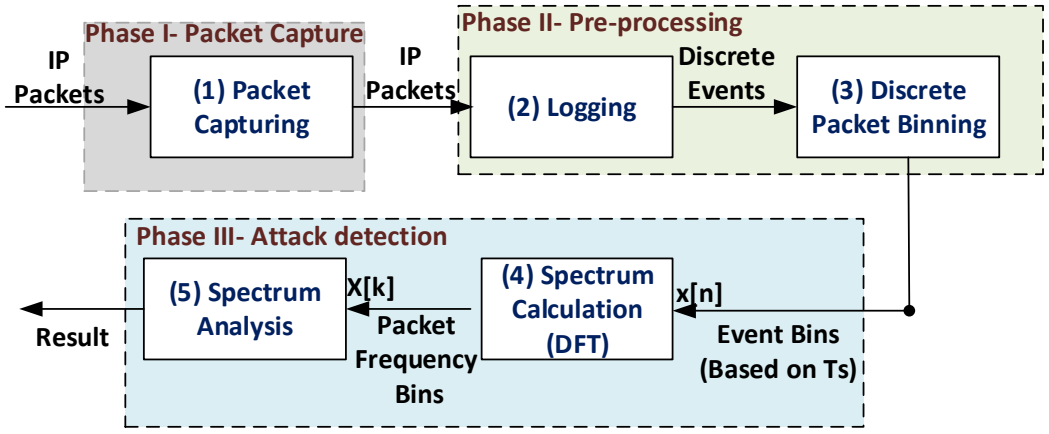


Figure 5.1: Proposed containerised attack detection model

5.3.1 Phase I-Packet Capture

During this phase, network traffic packets are captured for analysis using `tcpdump` (*TCPdump & LibPcap*, n.d.) or alternatives like Wireshark, Netflow collector, etc. For normal and DDoS traffic, we chose the Boğaziçi University Distributed Denial of Service dataset (Erhan & Anarim, 2020) and used `hping3` to launch a DoS attack from the attack container. Different instances of DoS and DDoS attack periods and attack frequencies (packet rates) were chosen to show the flexibility of the proposed method.

5.3.2 Phase II–Pre-Processing

This phase is composed of two steps: *Logging* and *Discrete packet binning*. We have implemented this phase in Python 3.8.

Logging: During this step, the server container maintained a real-time log of the captured traffic. The real-time log contains the number of packets, their arrival time (timestamp of each packet), the ports (sender and receiver), container ID and the IP addresses of the sender and receiver. We extracted only the samples of timestamp (first in $'sec'$ and then based on Nyquist Sampling rate (MARK, 2003)) from the maintained log. These timestamps are the discrete events of packets and used as an input to the discrete packet binning step. Note that samples can be produced for either all packet types, or for just a subset of the network traffic based on user-provided list of sender port(s), receiver port(s), container ID(s) and/or IP address(es).

Discrete packet binning: The *discrete packet binning* is an important step in pre-processing of the packets. For an efficient and inexpensive way of maintaining and processing the captured network traffic in resource-constrained devices, we employ a *binning approach*. Binning is a pre-processing technique to group discrete

events into contiguous, equally sized bins $bin[0] \dots bin[n-1]$ where n is the total number of bins. In our approach, packets are distributed into equal width bins. In other words, each bin's width is constant. We determine the total number of bins n (length of $bin[]$) as follows:

$$n = ceil(\frac{L_{end} - L_{start}}{t_s}) \tag{5.1}$$

where L_{end} and L_{start} represent the arrival time of the first and the last packet in the log file, respectively, t_s is the constant time interval representing the width of each bin, and $ceil()$ is a ceiling function.

The arrival time of each packet determines which bin it falls into. Thus, this binning method provides a discrete packet count ($bin[i]$ elements) over constant time intervals.

For illustration, we applied the binning method on the DoS traffic (Table. 5.1), which had 12500 packets in total. The packet rate for this traffic was 2500 packets/sec. The packets in the log file are distributed into $n = 5$ bins by using an Eq. 5.1, where each bin spans an interval of 1 sec and the arrival time of first and last packet was 57.598507 and 62.8165116 sec, respectively. For our sample traffic, we get:

$bin[]$ `= [2529 2530 2527 2488 2426]`

Each element $bin[i]$ $(0 \leq i \leq n-1)$ indicates the number of packets received in each 1 second wide bin. For example, from a total of 12500 packets, 2529 packets were received in the first $(i = 0)$ 1-second period and 2530 packets were received in the second $(i = 1)$ 1-second period. If no packets were received in any second, then a count of 0 would be assigned to that particular bin.

After the distribution of packets into their respective bins, the next step is to determine the estimated minimum Nyquist sampling frequency. The Nyquist sampling ensures that the information about the packets are fully preserved, and to get the accurate output minimum sampling frequency should be twice the maximum frequency of the packets(MARK, 2003). To calculate that minimum sampling frequency, we first find out the estimated maximum frequency of packets by determining the frequency of each bin as follow:

$$f_{bin}[i] = \frac{bin[i]}{t[i]} \tag{5.2}$$

where $(0 \leq i \leq n-1)$, array elements $f_{bin}[i]$ represent the frequency of each bin $bin[i]$ and $t[i]$ is the time difference between a start and end time of packets arrival in that particular bin.

By applying Eq. 5.2 on our sample bin array, we get the following frequency array f_{bin}:

$f_{bin}[]$ `= [2529 2531 2527 2489 2527]`

We obtain the maximum frequency f_{max} by:

$$f_{max} = max(f_{bin}[i]) \tag{5.3}$$

For our example, $f_{max}$is 2531Hz. Next, the estimated minimum Nyquist sampling frequency F_s is calculated by doubling f_{max}, that relates to the new sampling period T_s:

$$\begin{aligned} F_s &= 2 \times f_{max} \\ T_s &= 1/F_s \end{aligned} \tag{5.4}$$

For our sample, $F_s = 5062$Hz and $T_s = 19.75$ μs.

After determining the Nyquist sampling frequency, we re-distribute the packets in the log file into discrete bins of size equals T_s. The purpose of re-distribution at a higher frequency is to ensure that the network traffic is analysed thoroughly to detect the attacks accurately in full measure. Based on the calculated T_s and using Eq. 5.1 again, we yield the length $n^{'}$ to create the event bins (new array elements) $x[i]$, where $0 \leq i \leq n^{'} - 1$. This array of event bins is used as an input for the next phase.

For our sample, re-binning at a much finer resolution of $T_s = 19.75\mu$s results in the following bins:

```
x[n] = [ 1  0  0 ...  0  1 0 ]
```

5.3.3 Phase III–Attack Detection

DoS or DDoS attack detection has two steps: *spectrum calculation* and spectrum analysis.

Spectrum Calculation: In *spectrum calculation* step, an efficient algorithm, *Quick DFT* is used to calculate DFT and transforms the discrete time-domain data from the log file into discrete frequency samples with a subset of input. DFT converts the array x[] into a frequency-domain array X[] using the following formulae:

$$X[k] = \sum_{i=0}^{n-1} x[i] e^{-2j\pi ik/n} \tag{5.5}$$

where n is the total number of samples (size of x), k is the current frequency bin ($0 \leq k \leq n-1$), j is the imaginary unit $\sqrt{-1}$, and i is the current sample. It is important to mention here that DFT results include both positive and negative frequencies. The negative frequencies are complex conjugates of the positive frequencies and do not give useful information. So, we use a fast and optimised way to calculate only positive terms where the length of the transformed axis of the outcome is $n/2+1$. The outcome of the transform (amplitude) is mirrored about half of the sampling rate called the Nyquist frequency (Oppenheim & Schafer, 1998). We extract the amplitude spectrum by the following formulae: Note that $n = n/2+1$.

$$amp[k] = \frac{X[k]}{n} = \frac{\sqrt{Re(X[k])^2 + Img(X[k])^2)}}{n} \tag{5.6}$$

Applying Eq. 5.5 and Eq. 5.6, we convert our sample array $x[]$ into the following frequency domain array containing amplitudes:

```
amp[k]= [2.40858136e-15 6.04861884e-02
4.29612004e-02 ... 1.08618692e-01
1.61491802e-01 1.28808663e-01]
```

The amplitudes are mapped over the range of frequencies or frequency bins to analyse the packets in the frequency domain as follow:

$$f[k] = k\frac{F_s}{n} \tag{5.7}$$

The positive frequencies obtained for the example array are:

```
f[k]= [0.00000000e+00 2.01612261e-01 4.03
```

```
224522e-01 ... 6.32739919e+03]
```

Spectrum Analysis: In the final step, *spectrum analysis*, we use the network traffic analysis and spectrum calculations from phase 2 and the previous step to generate the *current frequency profile*, i.e., frequency and amplitude arrays, of a system in real-time. This current frequency profile is compared with the normal frequency signature (which incurs a linear cost) to detect DoS/DDoS attacks. Thus, in this step, characteristics of normal, DoS and DDoS spectrums are analysed.

Generally, our attack detection phase is highly efficient because Quick DFT has worst-case time complexity of $O(n\,log(n))$ and can be easily executed over resource-constrained devices that execute the containers of an ICPS application.

5.4 Experiments and Results

We modelled an attacker as an external container that discovers the target application by scanning for open ports using `nmap` and sends oversized packets to the application using `hping`(Vanney, 2021) command.

To determine the effectiveness of the proposed approach, we consider different five-second instances of normal traffic, DoS and DDoS traffics and categorised our experiments into *different packet rates*, to determine the flexibility of our proposed approach, and *different sampling frequencies*. The selection of right interval for detecting abnormal patterns in a transmitted data is a crucial task and should be chosen very carefully to detect the attacks accurately and within real-time. For our study, first we choose the interval of 1 sec as a heuristic, however, the bigger interval increases the chances of loosing the data. Therefore, we obtained the smaller intervals by following the Nyquist rule. But, lesser interval make

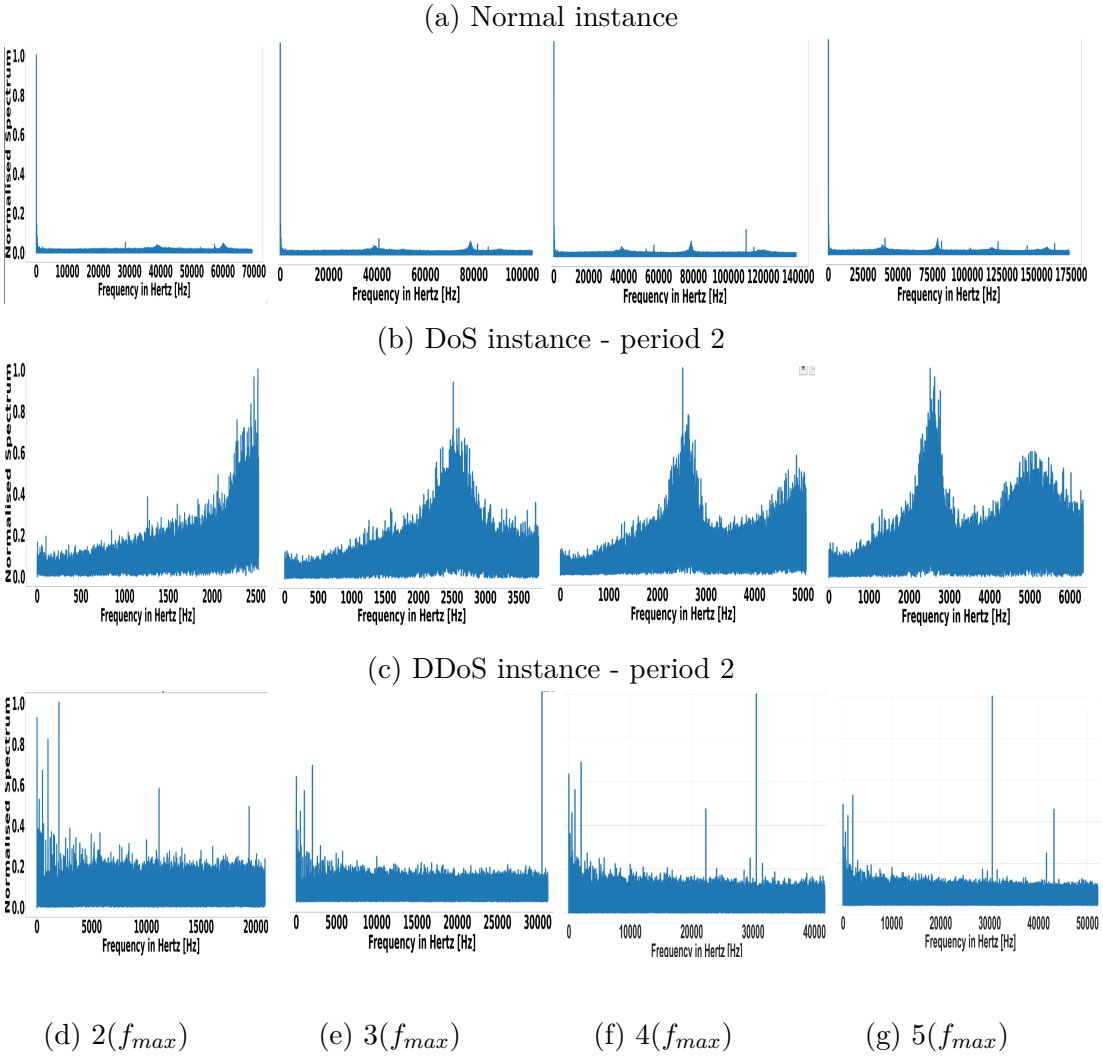


Figure 5.2: Spectrum of normal (a), DoS and DDoS instances with different packet rates (periods) (b,c,d,e) and sampling frequencies (f,g,h,i)

the system overloaded. As we are in need of light-weight detection approach for resource-constrained ICPS, we also monitor and compare the computational time of a proposed algorithm with different sampling frequencies in Table 5.1.

Fig. 5.2 shows the frequency signature of the normal traffic and its frequency signature under DoS and DDoS *flooding* attack. The figure presents two perspectives: different instances of traffic (Fig. 5.2a–5.2c) and different sampling frequencies (Fig. 5.2d–5.2g).

Table 5.1: Information about 5 sec instances of normal and attack periods and sampling frequencies with corresponding computational time in sec

Instances	**Period**	**Total Packets**	**Attack Frequency (Hz)**	**Packet start-time (sec)**	**Packet end-time (sec)**	f_{max}**in Hz**	**Computation time in sec**			
							2 (f_{max})	**3** (f_{max})	**4** (f_{max})	**5** (f_{max})
Normal		156227		10.011863	14.999871	69838	2758	7015	9015	10211
DoS	1	5000	1000	57.598507	62.816511	812	6.7	6.5	6.5	5.6
	2	12500	2500	79.392584	84.352585	2531	20.8	30.7	32	40.9
DDoS	1	51043	1000	95.00004	99.999832	22188	358	961.8	1285	886.3
	2	59976	1500	195.000063	199.99998	20832	851	565	1400	2243

Following the approach illustrated in Fig. 5.1, the frequency signature (Fig. 5.2a) and different sampling frequencies of the *normal traffic* were generated as a baseline. To generate the spectrum of the normal traffic, a total of 156227 packets were captured within 5 sec. To determine the *DoS attack*, we consider two different periods with attack frequencies of 1000Hz (1000 packets/sec) and 2500Hz (2500 packets/sec). During the first period, 5000 packets were captured, while for the period 2, 12500 packets were received in five seconds. Two different attack periods of *DDoS instance* were selected from the dataset for 5 sec. During the first period, a total of 51043 packets (1000 packets/sec) were captured; while 59976 packets (1500 packets/sec) were sent for another period. as shown in Table 5.1, but due to page limitations one instance is shown in Fig. 5.2.

The graphs (Fig. 5.2b and Fig. 5.2c) show that, when under attack, the frequency profile changes significantly from normal frequency signature and an attack can be signalled. In the case of the estimated minimum sampling frequency (Eq. 5.4), higher amplitudes (intensity/strength of attack) are found in lower frequencies for DDoS traffic. Whereas for DoS traffic, the highest amplitude is located in the higher frequencies band (Y. Chen & Hwang, 2007). Note that we are only interested in prominent peaks (amplitudes), because, usually, the amplitude of very high frequencies are harmonics or simply noise and contain unimportant

information. It was also observed that as the sampling frequencies increase, DoS attack signature patterns remain the same while the patterns changed for the DDoS frequency signatures.

5.4.1 Analysis and Insights

Frequency signature: The spectrums of normal traffic, shown in the graphs (Fig. 5.2a), exhibit uniform distribution along with all the frequencies. We analysed the attacks by comparing their frequency signature (in terms of dominant peak(s)) with the normal traffic spectrum. The DoS traffic pattern (Fig. 5.2b) shows the single dominant peak or single dominant frequency among different frequency bands. A single peak at 2500Hz indicates the attack at a particular frequency band. Whereas, for *DDoS* instance (Fig. 5.2c), the graph indicates that DDoS spectrums were also different from normal traffic and DDoS spectrum has at least two dominant peaks at different frequency bands.

Different sampling frequencies: We have analysed the normal and attack traffics on different sampling frequencies where the minimum and first sampling frequency was identified by using Nyquist Sampling Theorem. Then, we take 3, 4 and 5 times of $f_{max}$of each traffic instance (normal, DoS and DDoS traffics). The corresponding computation time in $'sec'$ was also calculated for the proposed algorithm by using each sampling frequency, shown in Table 5.1.

The maximum frequency (f_{max}) for normal traffic is 69383Hz. We analysed no significant change in the spectrum pattern of normal traffic by using all of the sampling frequencies as mentioned earlier. However, the computation time varies for each sampling frequency, as shown in Table 5.1. We concluded that twice of maximum frequency is the best sampling frequency for normal traffic. For *DoS*, $f_{max}$for period 1 and period 2 are 812Hz and 2531Hz, respectively. For

instance, Fig. 5.2d and Fig. 5.2e show that the dominant frequency, for period 1, at the higher bands was determined within 6.7sec and 6.5sec. With four times f_{max}(Fig. 5.2f), the peak becomes more dominant, which shows that this higher frequency (calculated within 6.5sec) gives more accurate information about the DoS attack. For *DDoS attack* period 1 having $f_{max}$of 22188Hz and period 2 with $f_{max}$of 20832Hz, we observed the fluctuations in dominant frequencies. At the twice of f_{max}(Fig. 5.2d), the dominant peaks were at lower frequency bands. Nonetheless, by increasing sampling frequencies, we noticed more sharp and dominant peaks at higher bands. It shows the arrival of more packets within shorter time periods. This also indicates that for effective detection of DDoS attack, we have to select more higher frequencies.

The above observations indicate that our light-weight active security approach is flexible in dealing with different packet rates, different frequency signatures and different sampling frequencies.

5.5 Conclusion

As industrial computing moves towards light-weight Docker-based containerisation, key communication security concerns become increasingly important. To meet the communication security needs of containerised ICPS, specially in resource-constrained environment, we have proposed a containerised attack detection model where light-weight active security approach is implemented as an algorithm. The experimental results based on different packet rates and sampling frequencies indicate that the proposed method is light-weight and flexible for the detection of the intrusion in a network traffic at low computational cost. Future areas include identifying thresholds for differentiate between normal and attack traffic and using real-time data streaming techniques and carrying out a qualitative analysis of our

proposed approach.

Chapter 6

Prelude - Manuscript 3

The following chapter is published as a conference paper in *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)* under the title of *DDoS Attacks on Smart Manufacturing Systems: A Cross-Domain Taxonomy and Attack Vectors* (Zahid et al., 2022a). The third research objective (Section 1.2.1)- to provide a complete overview of multi-vectors and cross-domain DoS/DDoS threat landscape within ICPS has been addressed in manuscript 3.

ICPS are increasingly connected to the Internet and are vulnerable to newer and scalable cyber attacks affecting their availability. The imminent and most reported attack to the availability of ICPS is DoS and its variation, like DDoS (many-to-one attack). These attacks impede the accessibility of the components and prevent the data and/or control distribution across the network. The types, methods, duration, and techniques to launch DDoS attacks have rapidly evolved. Research on cyber threats related to ICPS needs to explore the comprehensive landscape of DoS/DDoS attacks. These attacks are challenging to prevent entirely, so significant attention is given to their detection to avoid later compromises on the performance and availability of ICPS applications. This chapter aims to develop a

taxonomy and analyse various attack vectors to determine the comprehensive cross-domain threats landscape in ICPS, which have not been consider in the literature thus far. We classified at least fifty-two different (distributed) denial of service attacks as Endpoint and Network attacks. Endpoint (distributed) denial of service attacks are cyber-to-physical, physical-to-physical, physical-to-cyber, and cyber-to-cyber attacks launched to degrade or disrupt the availability of endpoint devices and applications and their services. Network DoS/DDoS attacks depict attacks on the communication layer intending to interrupt the transmission among endpoints or congest/block the network bandwidth. This article also introduced various Endpoint and Network DoS/DDoS attack sub-classes. Furthermore, various attack scenarios were demonstrated using the Fischertechnik Conveyor System case study to demonstrate the applicability of the proposed taxonomy.

Chapter 7

DDoS Attacks on Smart Manufacturing Systems: A Cross-Domain Taxonomy and Attack Vectors (Manuscript 3)

7.1 Abstract

Denial of Service is a significant availability threat in Industrial Cyber-Physical systems and smart manufacturing is not an exception. The types, methods, and duration of these attacks have been evolving rapidly and their number has increased dramatically, reaching a new record in history. In particular, digitisation of the manufacturing process and increased connectivity have created a battleground between product quality of service and threats associated with cross-domains and multi-vector attacks that affect the manufacturing system performance. The existing research on cyber-threats related to smart manufacturing system does not consider the comprehensive landscape of denial of service attacks. In this

study, we classify well-accepted (distributed) denial of service attacks according to a proposed taxonomy, focusing on both the multi-vector attacks and cross-domain attacks. Utilising the taxonomy, more than fifty different denial of service attacks on smart manufacturing system were classified in terms of Endpoint and Network (distributed) denial of service attacks. As an example, a Cyber-Physical Conveyor System was used to examine the proposed taxonomy.

7.2 Introduction

ICPS are complex, multi-disciplinary engineered systems that manage and perform industrial operations by integrating intelligent control, ubiquitous computing, and effective communication technologies (Kayan et al., 2021). The heterogeneous infrastructure, increased digitisation, advanced level technologies, and high connectivity among the ICPS components have improved the manufacturing time and quality. However, as they are now more connected and exposed, security attacks targeting the availability of ICPS components and their services are also rising. *The major threat to the availability in ICPS is DoS and its variation like DDoS (many-to-one attack)* that disrupts the access of some or all components by prohibiting the data and/or control distribution over the network (Zacchia Lun et al., 2019). ICPS are inherently resource-constrained and lack comprehensive security defenses, making them even more vulnerable to newer and varied DoS/DDoS attacks (Zahid, Kuo & Sinha, 2021a).

DDoS attack types, methods, duration, and techniques have been evolving rapidly. More recently, attacks have become more varied and numerous, but their duration and rate have decreased. Recent analyses of DDoS attacks (Alexander, Oleg & Yaroslav, 2022; Omer & Vivik, 2021) show that the Q4 '21 was the busiest quarter in 2021 for attackers, with 86,710 attacks registered on resources globally,

94% of them lasting less than 4 hours, and the *Manufacturing industry* experienced a 641% QoQ (Quarter-on-quarter) increase in the number of attacks. *Multi-vector attacks* are also a growing trend in which attackers combine many DDoS tactics into a single composite attack (Sean, 2022). These attacks are complex and require additional challenges for detection and prevention. The most frequent contributor to multi-vector attacks is generic UDP based DDoS attacks, while Simple Network Management Protocol (SNMP) is emerging as an important vector.

This study aims, due to the increasing trend of DDoS attacks in manufacturing and multi-vector attacks, to develop key classification criteria and analyze attack vectors as the way to identify the comprehensive cross-domain threats landscape in the *smart manufacturing system*, which have not been explored in the literature thus far.

This taxonomy can guide the planning and implementation of defenses in systems. We also demonstrate different attacks in the context of CPCS (Leitão, Barbosa, Funchal & Melo, 2020) to provide concrete use cases for each attack type.

The rest of the paper is organized as follow. Section 7.3 describes related works. Sections 7.4–7.5 present the proposed taxonomy and concrete examples of identified attack types. Section 7.5.6 reports conclusions and future work.

7.3 Related Work

Numerous CPS architectures have been proposed in literature (Ahmadi, Sodhro, Cherifi, Cheutet & Ouzrout, 2018; J. Lee, Bagheri & Kao, 2015; S. Huang et al., 2015; Jiang, 2018; J.-P. A. Yaacoub et al., 2020; Cao et al., 2020). For this study, we utilise the popular 3C architecture (Ahmadi et al., 2018). In this architecture, a CPS has three layers: *physical*, *communication* and *application* that respectively

relate to physical sensors and actuators, the transmission of data and control information between components, and high-level system functionality.

ICPS, in the context of smart manufacturing, provide flexible connection, interaction and synchronisation among different physical and cyber components across the different layers. The heterogeneous nature of these components, the increased connectivity, and the vulnerabilities in networks and platforms have expanded the attack surfaces. The effect of attacks on one layer or domain propagates to a cross-domain or cross-layer, which are not even attacked directly. Cross-domain cyber attacks are therefore critical in addition to traditional cyber attacks that affect the manufacturing industry. Some taxonomies for cyber-physical threats in smart manufacturing ((M. Wu & Moon, 2017, 2018)) have been presented, but none offer a comprehensive overview of DoS/DDoS attacks on cross-domains.

The research on DoS/DDoS attacks is extensive and several DoS/DDoS taxonomies, based on OSI layers, TCP/IP layers, IoT, ICS (Industrial Control Systems), cloud layers and WSN (Wireless Sensor Networks) have been presented in the literature ((Zeb, Baig & Asif, 2015; Salim, Rathore & Park, 2020; Mahjabin, Xiao, Sun & Jiang, 2017; Patani & Patel, 2017; Zargar, Joshi & Tipper, 2013; Bhardwaj, Subrahmanyam, Avasthi, Sastry & Goundar, 2016; Gavric & Simic, 2018)). Nonetheless, the existing studies are not in the context of smart manufacturing, do not provide a complete overview of DoS/DDoS threats at each CPS layer, and do not consider multi-vectors attacks and cross-domains attack. Furthermore, a denial of service attack can compromise wired communications, but no studies have addressed this in detail so far.

In this study, by identifying and characterizing all existing works, we have introduced an easy to understanding classification mechanism w.r.t cross-domain DoS/DDoS attacks on smart manufacturing systems. This classification can be

helpful for cyber security and manufacturing disciplines experts to understand the cross-domain attacks in smart manufacturing.

7.4 A Taxonomy of cross-domain DoS/DDoS attacks on smart manufacturing system

By combining availability attacks on the manufacturing process (M. Wu & Moon, 2018) and following the manufacturing ICPS architecture (Ahmadi et al., 2018), we have developed a taxonomy of cross-domain DoS/DDoS attacks on the smart manufacturing system. The availability attacks in the manufacturing process affect the availability of the equipment like sensors, actuators, and controllers. The equipment's attack can damage different physical components in manufacturing systems. ICPS, on the other hand, have sensors, actuators, controllers, networks and HMI as the basic components (Kayan et al., 2021). Therefore, we have used the term *Endpoint* in our taxonomy to categorise the attacks on the devices/applications in the physical and cyber layers. Thus, the cross-domain attacks on endpoint devices are cyber-to-physical, physical-to-physical, physical-to-cyber, cyber-to-cyber. Similarly, communication between endpoint devices is a critical component in the manufacturing process, even if they are located in different industries and in distant locations. These endpoint devices are mostly connected through wired and/or wireless infrastructure using industrial and network protocols interfaces. These infrastructures are responsible for transmitting important data for the factory, services, manufacturing, and users. Furthermore, within ICPS, real-time communication occurs when sensor readings or control commands are sent over wireless or wired communication links. Therefore, we have used the term *Network* to categorise the DoS/DDoS attacks on the communication layer.

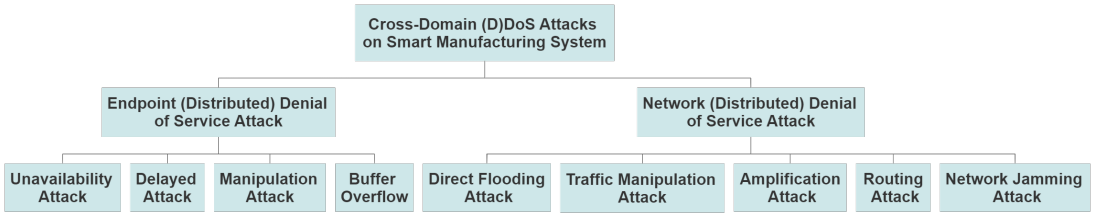


Figure 7.1: Taxonomy of cross-domain DoS/DDoS attacks on the smart manufacturing system

The taxonomy of cross-domain DoS/DDoS attacks on the smart manufacturing system is categorised on the *Endpoint (Distributed) Denial of Service Attack* and *Network (Distributed) Denial of Service Attack* classes, as shown in Fig. 7.1. The adversaries launch **Endpoint (Distributed) Denial of Service** attacks to disrupt or degrade the availability of equipment or their services. Whereas, an attacker conducts **Network (Distributed) Denial of Service** attacks to interfere with the communication among legitimate endpoints or congest/block the accessibility of network resources (network bandwidth).

The Endpoint and Network DoS/DDoS classes are further categorised into into sub-classes as shown in Table 7.1 and Table 7.2, respectively. This categorisation is based on the fact that an attack on each endpoint device is not interpreted as an attack on the equipment itself but could also be interpreted as an attack on the communication link between the sensing and the receiving endpoint devices (Fawzi, Tabuada & Diggavi, 2014). Each sub-class belongs to attacks with different types (whether an attack is DoS/DDoS), modes (direct by an attacker or indirect by malware/botnets), and vectors (methods to perform a specific attack).

This paper uses the case study Fischertechnik Conveyor System (CPCS) to demonstrate the developed cross-domain DoS/DDoS attacks taxonomy. This system is based on a sequence of modular conveyor belts that can be re-arranged to transfer parts from one place to another, constituting a CPS, as illustrated in Fig. 7.2.

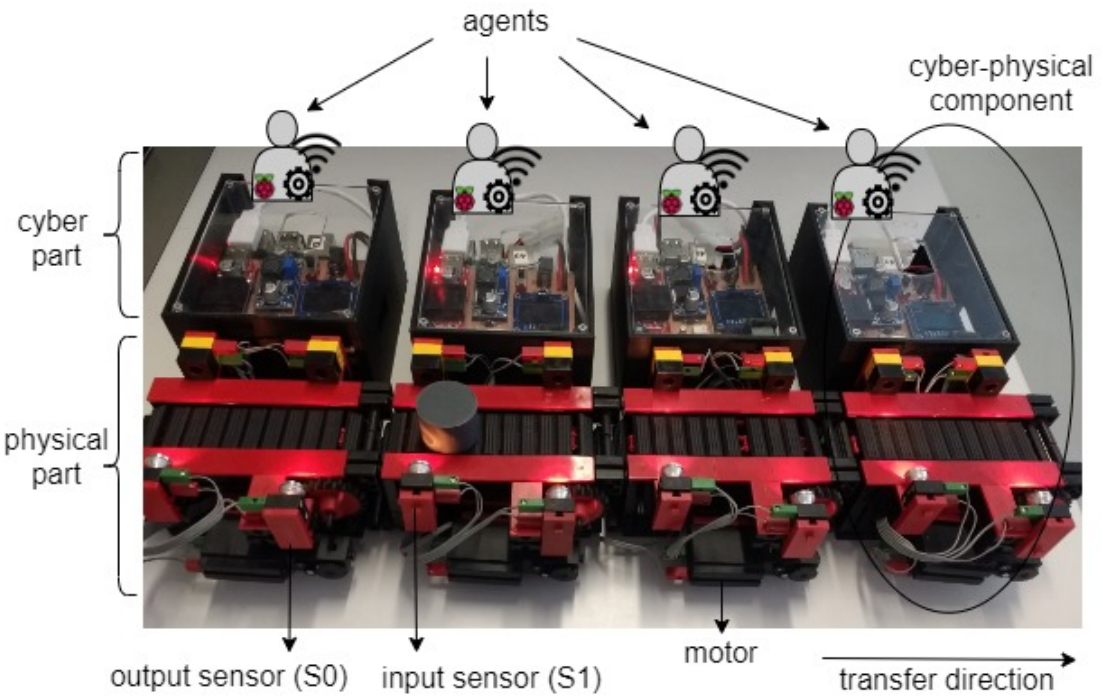


Figure 7.2: Cyber-physical conveyor system (Funchal et al., 2023)

According to the logical control perspective (Leitão et al., 2020), the different agents controlling the sequence of conveyor belts communicate by exchanging messages to synchronize the transfer of the parts among the conveyor belts. As an example, considering the first two conveyor modules of Fig. 7.2, the second conveyor only starts its motor when its agent receives the "tokenTransOut" message sent by the agent of the previous conveyor (meaning that the part is already at the end of the first conveyor belt and is detected by the S0 sensor). Similarly, the first conveyor belt only stops its motor when its agent receives the "tokenTransIn" message from the agent representing the posterior conveyor belt (the part arrives at the beginning of the conveyor and is detected by the S1 sensor).

The following section provides the concrete scenarios for each attack type (one from each sub-class) to demonstrate the applicability of the proposed taxonomy.

7.4.1 Endpoint DoS/DDoS Attacks on CPCS

This section presents the sub-classes of Endpoint DoS/DDoS attacks, and their corresponding attack scenarios on CPCS.

Table 7.1: Classification of Endpoint DoS/DDoS Attacks

Sub-Class	Names	Type	Mode	Vector(s)
Unavailability Attack	Permanent DoS (Gavric & Simic, 2018)	DoS	Indirect	damage firmware or upload inappropriate BIOS using Malware
	Land Attack (Mahjabin et al., 2017)	DDoS	Direct	use of source and destination IP addresses of a victim
	Tear Drop (Salim et al., 2020)	DoS	Direct	overlapped off set value, fragmented packets
	IP Packet Option (Zeb et al., 2015)	DDoS	Direct	random value for optional fields of an IP packet
	Ping of Death (Salim et al., 2020)	DDoS	Direct	Spoofed data packet having size bigger than maximum size of packet
	SIP[a] flood (Mahjabin et al., 2017)	DDoS	Indirect	SIP requests messages, SIP call control messages
	UDP[b] flooding (Patani & Patel, 2017)	DDoS	Direct	UDP Attack Packets, open ports, Spoofed IP address
	Denial of message Attack (McCune, Shi, Perrig & Reiter, 2005)	DDoS	Direct	use of benign network failures

Table 7.1: Classification of Endpoint DoS/DDoS Attacks (Continued)

Sub-Class	Names	Type	Mode	Vector(s)
Delayed Attack	PTP[c] Attack (Mizrahi, 2014)	DoS	Both	spoofed time protocol packets or control packets(Announce, Signal and management messages), use of configuration flaws, Rogue master attack
	Unfairness Attack (Gavric & Simic, 2018)	DoS	Direct	set "aMacBattLifeExt" to true.
	Sensor-Mac Attack (Quentin & Monnet, 2015)	DoS	Direct	announcement of multiple sleep delays and more virtual clusters that nearby nodes can store and handle in their internal tables, announcement that sync packet will sleep in T$syncmax$, use of MAC spoofing strategy
	Jamming attack on Time Synchronisation (Quentin & Monnet, 2015)	DoS	Direct	use of SYN packets, time synchronization outputs modifications, nullify preamble energy
Manipulation Attack	Data aggregation attack (Gavric & Simic, 2018)	DoS	Indirect	device configuration, device model, open ports, service information, threshold values information, runtime configurations, temporal features correlated with device physics and operations
	sensor data manipulation attack (Carvalho, Wu, Kwong & Lafortune, 2018; M. Wu & Moon, 2018)			
	Actuator data manipulation attack (Carvalho et al., 2018)			
	Controller logic and control command attack (M. Wu & Moon, 2018)			

Table 7.1: Classification of Endpoint DoS/DDoS Attacks (Continued)

Sub-Class	Names	Type	Mode	Vector(s)
Buffer Overflow	Denial of sleep (Gavric & Simic, 2018)	DoS	Direct	cause battery power exhaustion with inappropriate input
	Desynchronisation (Quentin & Monnet, 2015)	DoS	Direct	use of fake sequence or control flag for a message interception, use of spoofed MAC address and announcement of a new sync delay to some neighbor nodes
	TCP-SYN Attack (Mahjabin et al., 2017)	DDoS	Both	use of spoofed SYN packets, spoofed non-exisitng IP addresses or botnets
	Neptune Attack (Elleithy, Blagovic, Cheng & Sideleau, 2005)	DDoS	Direct	use of spoofed SYN packets on a specific port only
	PUSH-ACK (Zargar et al., 2013)	DDoS	Indirect	use of PUSH and ACK bit of TCP header
	Slowloris (Mahjabin et al., 2017)	DDoS	Direct	HTTP requests, open HTTP connections
	Slow Rate (Patani & Patel, 2017)	DDoS	Direct	connection establishment with valid HTTP request
	HTTP[d] fragmentation (Salim et al., 2020)	DDoS	Indirect	HTTP packets in the form of fragments, use of multiple connections
	Deceptive Jamming (Gavric & Simic, 2018)	DoS	Direct	transmission of regular legitimate packets
	R.U.D.Y [e] (Patani & Patel, 2017)	DDoS	Direct	exploits form submission field, use of multiple HTTP POST connections

[a] Session Initial Protocol, [b] User Datagram Protocol,[c] Precision Time Protocol Protocol,[d] Hyper Text Transfer Protocol, [e] Are.You.Dead.Yet

7.4.2 Unavailability Attack

Unavailability attacks are the disruptive attack strategies that an adversary can use to disable, deactivate, or crash the endpoint devices, making them unavailable (temporary or permanently) to perform their services. For example, an attack on a sensor makes it unavailable to sense the environment. Therefore, depending on the cross-domain attacks on endpoint devices, we have categorised this sub-class into further attacks, as shown in Table 7.1.

Scenario 1- ***User Datagram Protocol (UDP) Flooding Unavailability Attack*** *on Output Sensor (S0):* An attacker launches a UDP flood attack on the output sensor (i.e., S0) (Fig. 7.2) by sending a large number of IP packets containing UDP datagrams to the random ports on the output sensor. As a result, the output sensor is disabled to send "tokenTransOut" to the next conveyor (S1). Consequently, S1 will never receive the message to open the motor (shown in (Fig. 7.3)), and the motor will remain turned off forever.

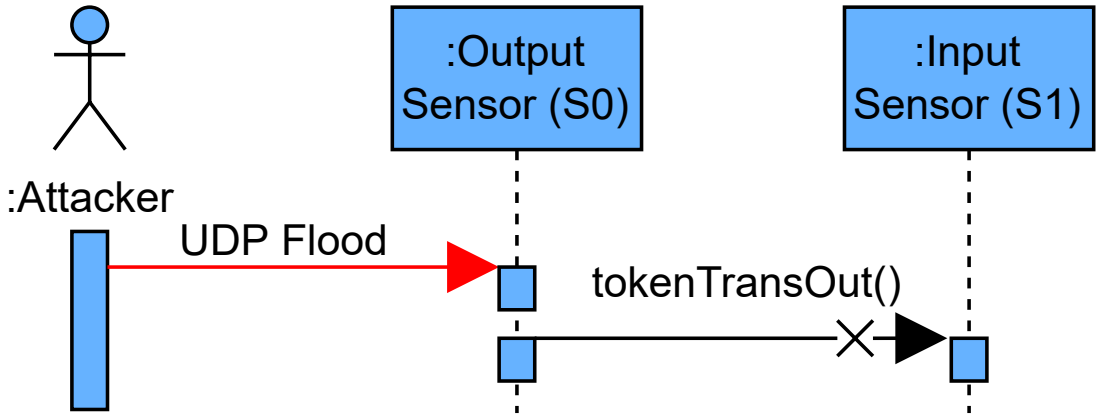


Figure 7.3: UDP flooding attack on Output Sensor (S0)

7.4.3 Delayed Attack

Delayed attacks involve the disruptive techniques an attacker uses to introduce the delays in timing or synchronisations of the endpoint devices. This attack is

further categorised based on the different attacks, as illustrated in Table 7.1.

Scenario 2- ***Sensor-MAC (S-MAC) Synchronisation Attack*** *on Input Sensor (S1):* S-Mac targets messages exchanged for synchronising the sleeping and listening schedules to transfer the objects on the conveyor belts. During the synchronisation process, an attacker acts as a synchroniser and sends a sync packets to all its neighboring agents, announcing that it will sleep in T$syncmax$ (maximum delayed value permitted by the protocol) and immediately goes to sleep. As a result, the attacker affects the synchronisation step of the sensor S1 (under attack) that has to wait for T$syncmax$ before starting its sleep period.

7.4.4 Manipulation Attack

The manipulation attacks (Table. 7.1) are the degradation of service attacks with an intention to produce the wrong output. The attackers exploit design or programming vulnerabilities in the endpoint devices or use HMI to manipulate (disable, enable, erasure, insert) their stored or run-time data by using malware.

Scenario 3- ***Actuator Data Manipulation Attack*** *on the Motor:* Attackers can attack the actuators' operations by overriding their control actions for a particular controllable event, for instance, enabling an event that is currently disabled and vice versa. Furthermore, malware can be used to change the actuator specifications, causing physical consequences; attackers can also alter maintenance schedules and processes, resulting in machines wearing out. For example, the motor remains in open mode if attackers maliciously modify the "tokenTransIn" message to "tokenTransOut message even though the agent (from the posterior conveyor belt) has sent "tokenTransIn" message to stop the motor.

7.4.5 Buffer Overflow

The buffer overflow is also a degradation of service attack which is used to target (waste or consume) the resources (CPU, memory, battery, disk space, ports, or resource-intensive features of user applications) of endpoint devices with an intention to corrupt their behaviors.

Scenario 4- ***Desynchronisation Attack*** *between S0 and S1*: Desynchronisation attack refers to the disconnection of an established connection (Gavric & Simic, 2018) that results in the waste of energy. For example, The malicious node (S0) sends regular requests for connection establishment to the sensor S1. These requests make the established connection desynchronised and both sensors can not communicate in a direct manner. In addition to that, additional power of S1 gets wasted in responding to the malicious node S0.

Table 7.2: Classification of Network DoS/DDoS Attacks

Sub-Class	Names	Type	Mode	Vector(s)
Direct Flooding Attack	MAC[a] flood (Zeb et al., 2015)	DDoS	Direct	use of invalid MAC addresses
	ICMP[b] flood (Bhardwaj et al., 2016)	DDoS	Both	ICMP_ECHO_REQUEST/REPLY packets, broadcast IP address, Spoofed source address
	Hello flood (Salim et al., 2020)	DDoS	Direct	Hello request to legitimate node with high power
	Flood Rushing (Kupwade Patil & Chen, 2013)	DDoS	Direct	use of existing routing path and sends continuous ROUTE-REQUEST to its neighbors
	SCTP[c] flood (Stewart, 2007)	DDoS	Direct	use of spoofed optimistic SCTP selective acknowledgement (SACK) packet
	QUIC[d] flood (Langley et al., 2017)	DDoS	Direct	QUIC handshake (Intial message)

Table 7.2: Classification of Network DoS/DDoS Attacks (Continued)

Sub-Class	Names	Type	Mode	Vector(s)
	Fraggle attack (Zargar et al., 2013)	DoS	Direct	UDP_ECHO packets, open ports, spoofed IP adresses
Traffic Manipulation Attack	HTTP[e] flood (Mahjabin et al., 2017)	DDoS	Direct	use of HTTP GET/POST requests
	UDP[f] Fragmentation Attack (Salim et al., 2020)	DDoS	Direct	use of larger attack packet size
	Profibus Attack (Kayan et al., 2021).	DoS	Direct	Lack of validation, amount of time Legitimate Profibus packets send, sending malicious LLDP (Link Layer Discovery Protocol) message
	Profinet Attack (Kayan et al., 2021).	DoS	Direct	Lack of validation, amount of time Legitimate Profinet packets send, sending malicious LLDP (Link Layer Discovery Protocol) message
	Modbus Attack (B. Chen, Pattanaik, Goulart, Butler-purry & Kundur, 2015).	DoS	Direct	Packet Length, Request/Response message containing malicious values for the data field option, function codes, False Modbus messages, configure sleep time
	HART[g] Attack (Eduard, n.d.)	DoS	Direct	packets malformed from field device to the DTM component
	EtherCat Attack (Granat, Höfken & Schuba, 2017)	DoS	Direct	Lack of validation, manipulated EtherCatframes , use of spoofed packets
	DeviceNEt Attack (Murvay & Groza, 2018)	DoS	Direct	exploit of session handle field, sending command packets to EtherNet/IP Web Server Module , use of TCP to keep large number of connection open, connection and device identities
Amplification Attack	DNS[h] Attack (Mahjabin et al., 2017)	DDoS	Both	request/sec to DNS server, spoofed IP addresses, sizeable resource record, message size,
	NTP[i] (Mizrahi, 2014), (Salim et al., 2020)	DDoS	Indirect	MON_GETLIST command to exploit NTP, spoofed source IP addresses, misconfiguration of clock
	QUIC (Langley et al., 2017)	DDoS	Both	spoof victim's IP address, and sending "hello" message,

Table 7.2: Classification of Network DoS/DDoS Attacks (Continued)

Sub-Class	Names	Type	Mode	Vector(s)
	DTLS[j] (Rescorla, 2022)	DDoS	Indirect	spoofed victim's address
	SNMP (M. Wu & Moon, 2018)	DDoS	Indirect	use of SNMP packets
	SSDP[k] (G. Singh & Singh, 2017)	DDoS	Indirect	use of spoofed UDP packets and spoofed IP address
Routing Attack	Blackhole Attack (Gavric & Simic, 2018)	DoS	Direct	use of null route (routing table entry), inactive host's IP address, or an IP address that has no assigned hosts, loopback address, and receiving RREQ message and replies using Route REPlY (RREP) message
	Sinkhole Attack (Kupwade Patil & Chen, 2013)	DoS	Direct	acquiring knowledge about in use routing protocol and advertisement of a fake routing update
	Sybil (Gavric & Simic, 2018)	DoS	Direct	malicious node makes multiple identities on same node (identities spooking)
	Wormhole Attack (Salim et al., 2020)	DoS	Direct	use of low latency link to tunnel packets from one point to other
	Selective forwarding (Salim et al., 2020)	DoS	Direct	use routing table to drop data/ACK packets or to reach base station
Network Jamming Attack	Constant Jamming (Gavric & Simic, 2018; Salim et al., 2020)	DoS	Direct	jammer may target the entire or a fraction of channel bandwidth, broadcast powerful signal all time
	Random Jamming (Kupwade Patil & Chen, 2013)	DoS	Direct	emitting periodic pulses of jamming signals
	Sporadic Jamming (Kupwade Patil & Chen, 2013)	DoS	Direct	changing a bit of data frame
	Reactive Jamming (Gavric & Simic, 2018; Quentin & Monnet, 2015)	DoS	Direct	use of interfering radio signal over the transmission of legitimate packets

[a] Media Access Control, [b] Internet Control Message Protocol,[c] Stream Control Transmission Protocol,[d] Quick UDP Internet Connections Protocol,[e] Hyper Text Transfer Protocol, [f] User Datagram Protocol,[g] Highway Addressable Remote Transducer Protocol,[h] Domain Name Server, [i] Network Time Protocol,[j] Datagram Transport Layer Security Protocol, [k] Simple Service Discovery Protocol

7.5 Network DoS/DDoS Attacks on CPCS

This section gives an overview of sub-classes of attacks belonging to Network DoS/DDoS class, and the corresponding scenarios of attacks on CPCS.

7.5.1 Direct Flooding Attack

During flood attacks, massive amounts of traffic are sent over networks to exhaust their bandwidth so that legitimate traffic can not be transmitted. Almost any network protocol may be used for flooding either by using it directly (direct network flood) or by exploiting its vulnerabilities. Different direct flooding attack techniques are given in Table 7.2.

Scenario 1- ***Flood Rushing Attack*** *on the Communication Between S0 and the Motor*: During the flood rushing attack, the attacker floods the routing path between S0 and motor by sending a huge volume of ROUTE-REQUEST, through the alternate path. As a result, the legitimate route gets discarded and the adversarial route gets adopted. In this way, an attacker can gain full control of the authenticated motor or sensor to cause physical damage or alter the motor or sensor data.

7.5.2 Traffic Manipulation Attack

Attackers interfere or manipulate the traffic either by exploiting the vulnerabilities in the communication protocols (wireless and/or wired) or by modifying the header or payload of the packets. Different traffic manipulation attack techniques are given in Table 7.2

Scenario 2- ***EtherCAT Traffic Manipulation Attack*** *on S1:* EtherCAT is a request/response industrial ethernet protocol and provides a way to connect the equipment involved in the manufacturing process (Kayan et al., 2021). Generally, these protocols lack the validation mechanisms, which become the source of an attack. In addition, attackers modify and forward the packets to the slave legitimate devices. For example, during an EtherCAT traffic manipulation attack, an attacker injects manipulated EtherCAT frames into the EtherCAT network, causing direct damage to the manufacturing process. The attacker performs this attack to manipulate the data measured by an Input Sensor (S1) which cannot detect the object's arrival and does not start the motor on time.

7.5.3 Amplification Attack

Attacker accesses a third party server called a reflector/amplifier by a spoofed source IP address of the victim and depletes the victim's network bandwidth by reflecting a large number of packets to it. Table. 7.2 shows different amplification attack techniques.

Scenario 3- ***Quick UDP Internet Connections Protocol (QUIC) Amplification Attack*** *on the Agent's Communication with S0:* The attacker spoofs the agent's IP address (controlling the S0 sensor) and requests the information from several servers or other sensors. Using QUIC protocol, the attacker sends an initial "hello" message to the servers to start the QUIC connection. As QUIC

includes both the UDP transport protocol and TLS encryption, the server includes its TLS certificate in its first reply to the client. Through the spoofing of IP addresses and sending hello messages to the servers, the victim agent receives a large amount of unwanted data and its bandwidth gets exhausted. As a result, that victim agent either becomes unable to control the sensor S0 for object detection or to pass the message to S1.

7.5.4 Routing Attack

The attacker launches an attack by a distribution of false routing information. Different kinds of routing attacks are available in the literature, as shown in Table 7.2.

Scenario 4- ***Sinkhole and Blackhole Attacks*** *on the Conveyors Communication:* A DoS attack on the conveyor system is performed by using sinkhole and blackhole attacks on the communication between two conveyors. First, the attacker makes a fake advertisement of high connection to attract the traffic containing the "tokenTransIn" message. After receiving that message, an attacker uses a blackhole attack mechanism to drop "tokenTransIn" message. As a result, the previous conveyor will not receive a message and the motor will remain in operation (i.e., does not stop even with the part already transferred).

7.5.5 Network Jamming Attack

Jamming attacks are among the most harmful attacks since they directly compromise the entire system. The attackers utilise powerful transmitters to jam specific spectrum bands and block the transmissions and receptions of the packets on any network in the affected area. Various kinds of Jamming attacks are illustrated in Table 7.2.

Scenario 5- ***Sporadic Jamming Attack*** *on the communication between S0 and the corresponding agent*: An attacker performs sporadic jamming on the communication between output sensor S0 and the corresponding agent (controlling S0). An adversary will alter one bit in a data frame and force the sensor to drop that data frame. Because of the channel interference, it becomes difficult for the victim agent to distinguish whether its signal band is intentionally jammed or not. Therefore, the agent increases his transmitting power, causing resources to deplete faster.

7.5.6 Conclusion

We presented a cross-domain DoS/DDoS attacks taxonomy in the ICPS context. Concrete examples of the various attack types have been presented in the context of CPCS, to show the applicability of the proposed taxonomy. The examples illustrated that attacks on one layer could impact the other layer(s) of CPS. Also, attackers can combine different DoS/DDoS tactics into a single composite attack. Thus, the proposed taxonomy and attack scenarios are helpful to cyber security personnel to understand the characteristics of attacks on the smart manufacturing systems and develop appropriate defense strategies.

Chapter 8

Prelude - Manuscript 4

The following chapter has been published as a journal paper in *IEEE Transactions on Industrial Informatics* under the title of *Actively Detecting Multi-Scale Flooding Attacks & Attack Volumes in Resource-Constrained ICPS*. Overall, manuscript 4 addresses the fourth and fifth research objectives by designing, developing, and implementing the light-weight active security solution for resource-constrained ICPS (Section 1.2.1).

Resource-constrained ICPS are important but vulnerable critical infrastructure components under cyber attacks like DDoS flooding attacks. With massive traffic requests, these attacks overwhelm the limited resources of resource-constrained ICPS. Flooding attacks can be multi-scale, and their intime detection is critical as they can lead to crashes, network congestion, or performance degradation of the devices. To actively secure resource-constrained ICPS, there is a need for light-weight framework which can detect multi-scale flooding attacks and attack volumes by focusing on the frequency signature of the incoming traffic. In this paper, we have extended our work (Chapter. 4) and proposed a novel two-phase technique for spectrum analysis. The first phase detects the presence of attacks by using statistically robust and low computational overhead methods such as the

spectra similarity method and modified fast-entropy method. The next phase will determine the attack volumes based on these two methods combined *true* output. The experimental results and analysis show the outperforming performance of the proposed framework in terms of CPU and memory overhead and attack detection time compared to the existing state-of-the-art.

Using our proposed flooding technique, we also validated the newly created dataset discussed in Appendix. A. The dataset was created for intrusion detection employing a self-organised conveyor system controlled by multi-agent technology.

Chapter 9

Actively Detecting Multi-Scale Flooding Attacks & Attack Volumes in Resource-Constrained ICPS (Manuscript 4)

9.1 Abstract

The significant growth in modern communication technologies has led to an increase in zero-day vulnerabilities that degrade the performance of ICPS. DDoS attacks are one such threat that overwhelms a target with floods of packets, posing a severe risk to the normal operations of the ICPS. Current solutions to detect DDoS attacks are unsuitable for resource-constrained ICPS. This study aims to actively detect multi-scale flooding DDoS attacks and attack volumes on resource-constrained ICPS by analysing a magnitude spectrum of incoming traffic in a frequency domain. A two-phased technique is proposed for the spectrum analysis: detecting the attack presence and detecting the attack volume. Both

phases use a novel combination of light-weight and theoretically sound statistical methods. The effectiveness of the proposed technique is evaluated in terms of true and false positive rate, accuracy, and precision using BOUN DDoS 2020 and CICDDoS 2019 datasets. The proposed approach was implemented on a PLC-based system and demonstrated outperforming performance in terms of CPU and memory overhead, as well as attack detection time when compared to the existing state-of-the-art.

9.2 Introduction

Distributed Denial of Service (DDoS) is an immerse threat that disrupts or degrades some or all of the resources of *Industrial* Cyber-Physical Systems (ICPS) by preventing information distribution over networks (Agrawal & Kumar, 2022),(D. Zhang, Wang, Feng, Shi & Vasilakos, 2021). ICPS are highly distributed systems with numerous components interacting with each other and the physical environment. Inherently, ICPS, like PLCs, are resource-constrained with limited bandwidth, processing capacity, memory, and security capabilities (Zahid, Kuo & Sinha, 2021a). DDoS attacks exploit such limitations or vulnerabilities and cause delays, bandwidth depletion, buffer overflows, and crashes (Bhatia, Behal, Ahmed, Somani & Poovendran, 2018). DDoS attacks are categorised based on attack rate (High-rate or flooding attack and Low-rate DDoS) or their impact (resource-depletion or bandwidth depletion attacks) (D. Zhang et al., 2021) or in terms of Endpoint (devices/applications in the physical and cyber layers) and Network (communication layer) of ICPS (Zahid et al., 2022b). A recent analysis of DDoS attack trends shows that these attacks are evolving rapidly in terms of their types, volume (increased by 89% QoQ (Quarter-on-quarter)), and rates, which have increased dramatically (Omer, n.d.). The in-time detection of DDoS attacks,

particularly the multi-scale flooding attacks is critical and challenging due to the high operational destructive impact of flooding attacks on performance and safety (David & Thomas, 2019; Zahid et al., 2022b). "Multi-scale" refers to varying aspects of the attack traffic, including predictable and unpredictable attack rates, attack periods, attack densities, peaks, and attack volume (magnitude/ sheer number of attacks over a given time-interval-attack intensity) (Omer, n.d.; He et al., 2005).

This work explores the following research questions:

RQ1 What are the existing DDoS flooding attack detection strategies suitable for ICPS applications?

RQ2 What are the limitations of the works identified in RQ1 in resource-constrained ICPS applications?

RQ3 How can a light-weight active security technique be provided in resource-constrained ICPS to deal with multi-scale flooding attacks and attack volumes?

RQ4 How can the technique proposed in answering RQ3 be evaluated for its feasibility and effectiveness with respect to the works identified in RQ1?

We answer RQ1 and RQ2 through a literature survey presented in Sec. 9.3. RQ3 and RQ4 are answered (Sec. 9.4-Sec. 9.6) through an adapted Design Science research methodology to build and test the light-weight technique for detecting multi-scale flooding attacks in resource-constrained ICPS applications.

Our *survey results (Sec.9.3) indicate that the optimal solution for detecting multi-scale flooding attacks in resource-constrained ICPS is still an open research problem* (David & Thomas, 2020; Agrawal & Kumar, 2022). Existing works use anomaly-based, signature-based, or hybrid-based detection approaches (Tan,

Guerrero, Xie, Han & Vasquez, 2020; D. Zhang et al., 2021). These security approaches are heavy-weight, not memory-efficient, require significant processing capacity, and have high false positive rates. These limitations make the existing works ineffective, difficult to determine the flooding attacks in time, and expensive to develop and maintain (Tsobdjou, Pierre & Quintero, 2022), especially in resource-constrained ICPS (Agrawal & Kumar, 2022). Moreover, attack volumes comprehensively assess the number of attacks and their intensity. A higher attack volume typically implies a more significant threat. Thus, *resource-constrained ICPS require light-weight mechanisms to actively (dynamically and programmability) detect multi-scale flooding attacks and volumes early from the incoming traffic flow* (Zahid, Kuo & Sinha, 2021a).

This study aims to propose a light-weight active security technique to detect multi-scale flooding attacks and volumes in resource-constrained ICPS to address the identified issues. The proposed technique analyses the network traffic in the frequency domain and is based on straightforward spectral analysis and statistical approaches, which allow fast and accurate detection of DDoS attacks (Nooribakhsh & Mollamotalebi, 2020). Frequency domain analysis is a promising approach that provides a better understanding of network traffic which is often not possible via time domain analysis and enables accurate and robust attack detection using unique frequency signature of abnormal behaviour (He et al., 2005).

PLCs are industrially-hardened but resource-constrained computers on which ICPS software is deployed to control physical processes. DDoS attacks can very quickly overwhelm PLCs, causing substantial disruptions. Protecting PLCs from DDoS is challenging and has not received significant research attention (Verma et al., 2023). To the best of our knowledge, no existing work provides flooding attack detection within a PLC-based ICPS. *Another significant contribution* of this work is the implementation of our proposed attack detection technique on

the PLC-based ICPS.

To pursue the research questions, the primary contributions of this study are:

1. A systematic literature review of the existing DDoS flooding attack detection strategies (Section. 9.3).

2. This article proposes a novel two-phased light-weight active security technique to dynamically detect multi- scale flooding attacks and attack volumes in resource-constrained ICPS (Section. 9.4).

3. A novel multi-method approach to dynamically detect the presence of flooding attacks in PLC-based ICPS (Section. 9.4).

4. An experimental validation of our proposed approach on publicly available datasets: Boğaziçi University Dis- tributed Denial of Service (BOUN DDoS) dataset (Erhan & Anarim, 2020) and CICDDoS 2019 (Sharafaldin, Lashkari, Hakak & Ghorbani, 2019) (Section. 9.6).

5. The performance (memory and CPU overhead) evaluation of the proposed approach using PLC (Section. 9.6).

9.3 Related Works

Traditional security mechanisms like firewalls, routers or load balancers are ineffective in resource-constrained ICPS (Agrawal & Kumar, 2022; Verma et al., 2023) as they do not prioritise memory efficiency or resource optimisation. These mechanisms rely on prior knowledge of attack signatures (specialised patterns signaling threats); consequently, new and unseen attacks go unnoticed (Verma et al., 2023). Various DDoS attack detection surveys exist (Tan et al., 2020;

Table 9.1: State-of-the-art DDoS flooding attacks detection strategies based on Detection methods, Detection features, Attack volume (Yes/No), Domain analysis (Time/Frequency), Resource-constrained (Yes/No), Threshold (Static/Dynamic/Not-Applicable (NA)), Datasets used (Name/No (dataset not used))

References	**Detection Method(s)**	**Detection feature(s)**	**Vol**[2]	**Dom**[3]	**Res**[4]	**Th**[5]	**Datasets used**
(Ma, Almutairi, Alwakeel & Alhameed, 2023)	*Hierarchical Bayesian Network*	*Src*[6]*, Dest IP*[7]*, P*[8]*, service types, duration*	*No*	*T*[9]	*No*	*NA*	*NSL-KDD*
(Sivamohan et al., 2023)	*Krill herd optimisation, bi-LSTM*	*flow and connection data*	*No*	*T*	*No*	*NA*	*NSL-KDD*
(B. Liu et al., 2022)	Matrix multiplication	control data	No	T	No	NA	No
(Tsobdjou et al., 2022)	Shannon Entropy	Src, Dest IP	No	T	No	D	No
(Zahid, Kuo & Sinha, 2021a)	QuickDFT	Timestamp	No	Freq[10]	Yes	NA	BOUN DDoS 2020
(B. Li et al., 2021)	*Federated Learning, neural network*	*flow data*	*No*	*T*	*No*	*S*	*No*
(Ali et al., 2021)	Entropy, Sequential probability ratio	Dest IP	No	T	No	S[11]	DARPA98, DARPA2000, CICDDoS 2019
(David & Thomas, 2020)	Renyi Entropy, Stochastic gradient	Src, Dest IP, Src, Dest ports, P, packet size	No	T	No	D[12]	MIT98, CAIDA2007
(Kordestani et al., 2020)	Fuzzy logic, neural network	flow rate, pressure	No	T	No	NA	No
(Mahmoud et al., 2020)	Random Conditional Probability	Control data	No	T	No	S	No
(Raiyat Aliabadi, Seltzer, Vahidi-Asl & Ghavamizadeh, 2021)	Bayesian-based search, score technique	not mention	No	T	Yes	NA	No
(David & Thomas, 2019)	Mean, variance	Src, Dest IP, Src, Dest ports, P	No	T	No	D	DARPA2000, DARPA98
(Fouladi, Ermis & Anarim, 2019)	DFT, Sparse representation model	Timestamp	No	Freq	No	S	CAID2011
(Behal, Kumar & Sachdeva, 2018)	Generalised Entropy, Information Distance	Src IP	No	T	No	S	MIT98, CAIDA2007, FIFA98
(K. Singh, Dhindsa & Bhushan, 2018)	Shannon Entropy	Src, Dest IP, Src, Dest ports, P	No	T	No	D	No
Our work	*Fast-entropy, Jaccard similarity, Euclidean distance*	*Timestamp*	*Yes*	*Freq*	*Yes*	*D*	*CICDDoS 2019, BOUN DDoS 2020*

[1]References, [2]Attack volume, [3]Domain analysis, [4]Resource-constrained, [5]Threshold, [6]Source, [7]Destination IP, [8]Protocol, [9]Time, [10]Frequency, [11]Static, [12]Dynamic

Zahid et al., 2022b; Bhatia et al., 2018; D. Zhang et al., 2021), but none consider resource-constrained ICPS.

Our survey includes generic and ICPS-specific techniques for detecting flooding attacks. Table 9.1 shows a comparison of various state-of-the-art DDoS detection strategies.

Several works employ information entropy, statistical dispersion, similarity/dissimilarity, machine learning, hybrid and knowledge-based methods for attack detection. Information entropy-based detection mechanisms measure changes in the randomness of network traffic. However, most techniques are not ICPS-relevant, have low detection rates and/or have high computational costs (David & Thomas, 2020). Statistical dispersion methods become computationally time and resource-intensive when large numbers of network traffic detection features (inspection and analysis of both header and payload of packets) are considered (Nooribakhsh & Mollamotalebi, 2020). Using static thresholds is computationally inexpensive, but such techniques cannot adapt to changing traffic characteristics and are ineffective in dealing with new attacks (Tsobdjou et al., 2022). The spectral analysis can detect trends and irregularities in network traffic effectively (He et al., 2005),(Fouladi et al., 2019), but it has not been used in resource-constrained ICPS.

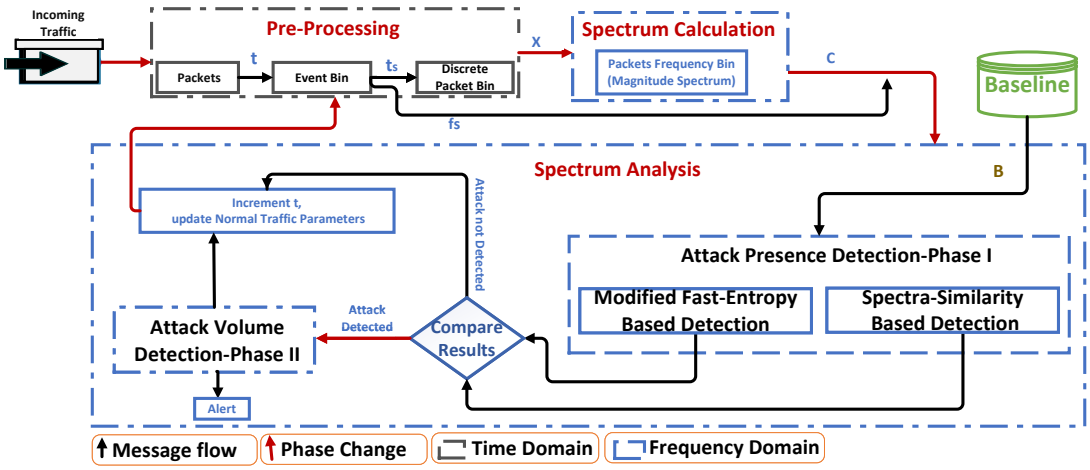


Figure 9.1: Proposed Light-Weight Multi-scale Flooding Attacks and Attack Volumes Detection Model

Existing techniques for detecting DDoS attacks in ICPS range from machine learning, signature-based, and knowledge-based approaches (B. Li et al., 2021; Zahid, Kuo & Sinha, 2021a; B. Liu et al., 2022; Mahmoud et al., 2020; Kordestani et al., 2020; Raiyat Aliabadi et al., 2021). These techniques require baseline information like historical data, models, forecasts or rules, and dictionaries of known attacks. These methods are not well-suited for resource-constrained ICPS because of high resource usage, and inability to detect previously unknown attacks. Another work (Zahid, Kuo & Sinha, 2021a) employs frequency analysis for DDoS attack detection in ICPS; however, detection relies on human observation and attack detection time is high.

Generally, most current works are validated against outdated datasets, making them ineffective against more sophisticated flooding attacks (Tsobdjou et al., 2022). Also, many of the research efforts use time domain analysis of network traffic and cannot detect patterns hidden in a noisy environment (He et al., 2005). Moreover, researchers proposed a single method of detection that may have limited effectiveness in a changing environment. Table. 11.1 shows that most works detect the presence of attacks in malicious traffic, but they do not identify the attack volume. No current work optimises performance overheads and consequently feature high attack detection times.

Overall, in contrast to the existing works, our work introduces a multi-method, light-weight, frequency domain technique to actively secure resource-constrained ICPS and uses a number of optimisations to further reduce resource consumption.

9.4 Proposed Light-Weight Multi-Scale Flooding Attacks and Attack Volumes Detection Technique

We propose a light-weight technique that actively detects multi-scale flooding attacks and attack volume on resource-constrained ICPS in the frequency domain. The proposed attack detection technique comprises *three major components: pre-processing, spectrum calculation, and spectrum analysis*, as illustrated in Fig. 11.2.

9.4.1 Pre-processing and Spectrum Calculation

Pre-processing is a significant step in attack detection. It involves the transformation of raw network traffic into a meaningful form for accurate analysis of malicious behavior or to uncover the hidden patterns or trends in the traffic.

This work employs a computationally efficient *binning method* for traffic pre-processing. The binning method ensures that the traffic is appropriately processed for accurate analysis in later steps of the proposed techniques. Binning involves extracting discrete packets from the network over a time window and distributing them into contiguous and equally sized event bins. The flooding attacks exhibit specific temporal traits; thus, binning methods help to identify the deviations or trends over different time intervals.

During traffic pre-processing, the first and last packet arrival times (t_{first} and t_{last}) and the total number of packets captured (p_{total}) are used to create equally sized bins x_i with the duration t_s (dynamic sampling time-period). $X = [x_1, x_2, ...x_n]$ is the array of all bins over a time-window t_w. The dynamic sampling time-period is computed by $t_s = 1/(2 \times f_s)$ where $f_s = p_{total}/(t_{last} - t_{first})$. The $2 \times f_s$ is used as the Nyquist sampling frequency. Each bin's duration is one t_s

and stores a count of the number of packets within the bin's time period.

Spectrum calculation uses a light-weight Quick Discrete Fourier Transform (QuickDFT) approach to transform the bins of packets counts in the time domain into the frequency domain (Zahid, Kuo & Sinha, 2021a). $C = QuickDFT(X)$ where $C = [c_1, c_2, ..., c_n]$ is the magnitude array of the frequency domain representation of the packets over the time-interval t_w (Zahid, Kuo & Sinha, 2021a).

Baseline data: Our detection technique detects anomalies or deviations from a baseline array ($B = [b_1, b_2, ..., b_n]$), representing the normal traffic where the system is not under attack. The baseline array is an iteratively updated array storing the magnitude values of baseline traffic over the frequency domain. Further detail about baseline data is in Sec. 9.5.1.

9.4.2 Spectrum Analysis

The spectrum analysis component consists of two phases: *attack presence* and *attack volume detection*. Attack presence detection is further divided into two sub-phases, shown in Fig. 11.2. *Spectra-similarity based detection* uses the *Jaccard Similarity metric* (Cha, 2007) to compare the similarity between normal and incoming traffic magnitude spectra with high accuracy. Jaccard similarity was selected due to its lower sensitivity to noise and based on the higher detection accuracy after extensive experimentation over various metrics, including Intersection, Kulczynski, Soergel, Canberra, Czekanowski, Tanimoto, and Hellinger (Cha, 2007). *Modified fast-entropy based detection* uses the *fast-entropy method* (David & Thomas, 2015) to monitor changes in entropy (uncertainty) of the magnitude spectrum of traffic through flooding attacks. Utilising the multi-method approach in phase-I improves the effectiveness of flooding attacks detection by providing distinct abnormal traffic behaviour and increasing the confidence level in detection

results. Both proposed methods contribute to identifying the flooding attack by analysing the magnitude spectrum in the frequency domain, but their applications differ. Jaccard similarity indicates the potential attack based on the deviation of incoming traffic from the established baseline. While, the modified fast-entropy detection focuses on the randomness of incoming traffic during the attack based on the calculated fast-entropy value.

The second phase, *volume detection*, uses Euclidean distance (a dissimilarity metric) as the primary measure (Cha, 2007). Euclidean distance is computationally simple and can analyse the correlation information more efficiently and effectively, even over a few features(Cha, 2007). The following analysis is all performed over the frequency domain.

Spectra-similarity based detection: The Jaccard similarity is computed over the baseline B and incoming traffic spectra C using Eq. 9.1 where $n = max(|C|, |B|)$ (Cha, 2007):

$$J = \frac{\sum_{k=1}^{n} c_k.b_k}{\sum_{k=1}^{n}(c_k)^2 + \sum_{k=1}^{n}(b_k)^2 - \sum_{k=1}^{n} c_k.b_k} \tag{9.1}$$

Trailing zeros are padded to ensure the two arrays are of the same size. Perfect similarity is indicated by 1, while 0 means no similarity (Cha, 2007). *An attack is present if the value of J lies within the range of 0 to 0.5.* The intention behind establishing this range is to provide a clear distinction between normal and malicious behaviour and effectively capture the deviation of incoming network traffic from the baseline.

Modified Fast-Entropy Method: The entropy value describes the dispersion or concentration of network traffic features (David & Thomas, 2020). The higher the concentration, the lower will be the entropy value. In flooding attacks, the number of packets increases significantly for a certain time-period, which results

in a dominant peak that indicates a dramatic fall in fast-entropy value.

To reduce the computation time, we first create a sub-array of C, denoted as Z, for the values greater than or equal to a lower control threshold β (calculated by using Eq. 9.8 discussed in Sec. 9.5.2). Let $Z = [z_1, z_2, \ldots, z_k]$ where $\forall z \geq \beta$ be the sub-array of C.

The fast-entropy values $H = [h_1, h_2, ...h_k]$ for each corresponding magnitude value in Z is calculated by Eq. 9.2 where h_j is the fast-entropy value of z_j.

$$\begin{aligned} & h_j = -log\frac{z_j}{\sum_{i=1}^{|Z|} z_i} + \lambda_j \\ & where \\ & \lambda_j = \begin{cases} |log\frac{z_j}{z_j+z_{j+1}}|, & \text{if } z_j \geq z_{j+1} \\ |log\frac{z_j+z_{j+1}}{z_j}|, & \text{if } z_j < z_{j+1} \end{cases} \end{aligned} \tag{9.2}$$

λ_j is the fast-entropy calibration factor based on (David & Thomas, 2015). In our study, to increase the accuracy of the multi-scale attack detection, we have modified the existing fast-entropy calibration factor (λ_j) empirically using Shannon's fundamental properties of information content (Cover Thomas & Thomas Joy, 1991). Several measures were trialled before Eq.9.2 was chosen as it exhibited the best accuracy. *The modified fast-entropy method detects an attack if incoming traffic's fast-entropy rate (δ_Z) is less than the flooding attack confirmation threshold (Θ).* Generally, the fast-entropy rate (δ) is the average of all entropy values H (K. Singh et al., 2018).

$$\delta = \frac{\sum_{k=1}^{|H|} h_k}{|H|} \tag{9.3}$$

An attack is detected if δ_Z is less than a threshold Θ (see Eq. 9.9 in Sec. 9.5.2).

Attack Volume Detection: Fig. 11.2 shows that the attack volume detection

phase triggers based on the *true* output of the attack presence detection phase. Euclidean distance is used to determine the attack volume, i.e., one-time extreme peak (one dominant peak) or peak volume (multiple dominant peaks) by using an Eq. 9.4.

$$d = |a_1 - a_2| \tag{9.4}$$

where a_1 and a_2 are two points on the real line in one dimension (either x-axis or y-axis). As we are only interested in the dominant peaks, we created a subarray called $\hat{H}$ from H where the values are above the upper control threshold (ξ) (calculated using Eq. 9.8 in Sec. 9.5.2). Let $\hat{H} = [\hat{h}_1, \hat{h}_2, ...\hat{h}_m]$ where $\forall \hat{h}_j \geq \xi$.

The attack volume is calculated with the Euclidean distance d between each fast-entropy value in $\hat{H}$ and the dominant peak's fast-entropy value (ζ_Z) where ζ_Z=$min(H)$. The peak volume is identified if the d values lie within the range from 0 to less than 0.5. The distance 0 means the peaks are close (have the same amplitude). An alert will be generated to give the information about the identified dominant peaks responsible for an attack. Equation 9.5 is performed for all values in $\hat{H}$.

$$alert = |\zeta_Z - \hat{h}_j| < 0.5 \tag{9.5}$$

9.5 Initialisation of Baseline, parameters and Thresholds Computation

This section discusses the methods to set the baseline and thresholds.

9.5.1 Initialisation of Baseline and parameters

The baseline is built using normal behaviours identified by the system when the usual messages are being sent between devices. First, a list of magnitude arrays

identified as normal traffic are collected, representing a range of normal traffics across different time intervals. The mean of each magnitude array is calculated, and the array with the maximum mean was selected as baseline (B). The maximum mean demonstrates the highest value (number of packets/magnitude) for the legitimate traffic at a particular destination IP. Fig. 9.2(a) shows the illustration of the baseline.

This baseline magnitude array is used to calculate the various parameters, such as baseline fast-entropy values (H_B) using Eq. 9.2, baseline fast-entropy rate δ_B using Eq.9.3 and baseline standard deviation σ_B that will be used to compute the flooding attack confirmation threshold using Eq.9.9 (Sec. 9.5.2).

The baseline is updated when no attack is detected, and the mean of the current magnitude array C is greater than that of the baseline. The baseline is then updated as $B = C$. The various parameters are also subsequently updated.

9.5.2 Thresholds Computation

For accurate attack detection, *threshold(s) identification and setting the values* are challenging tasks (K. Singh et al., 2018). The thresholds used in this study are dynamic (change with the changes in network traffic flow) and determined by the traffic distribution.

Our approach uses a *lower control threshold (β), and flooding attack confirmation threshold (Θ)* at phase-I (during modified fast-entropy based detection). To determine the peak volumes at phase-II, the *upper control threshold (ξ)* is identified. In essence, the amplitude of very high frequencies are the harmonics or noise which contains unimportant information (Zahid, Kuo & Sinha, 2021a); therefore, we created β to remove unimportant information (noise/harmonics) and to eliminate the chances of false positives and negatives. The purpose of establishing

an upper control threshold (ξ) is to identify attack volumes in resource-constrained environments while minimising the number of computations.

As our purpose is to make a robust and simple approach for resource-constrained ICPS, we have used Chebyshev's Inequality (Amidan, Ferryman & Cooley, 2005) to determine the upper and lower control thresholds. Chebyshev's Inequality is a valuable and flexible approach to assessing the proportion of observations that fall within a z (integer) range of standard deviations, as shown in the equation (Eq. 9.6) (Amidan et al., 2005).

$$[\mu - z.\sigma, \mu + z.\sigma] \tag{9.6}$$

During the flooding attack, the attack spectrum has higher amplitudes than the legitimate spectrum because attack network traffic contains more packets (Zahid, Kuo & Sinha, 2021a). Therefore, we have used the upper bounds of the interval both for the lower control (β) and upper control (ξ) thresholds. In addition, to overcome the issue of false alarm rates, we have also introduced the threshold calibration factor called Freeman's index (v) with Chebyshev inequality. This factor is calculated by Eq. 9.7.

$$v = \frac{|1 - f_s|}{|C|} \tag{9.7}$$

Thus, β and ξ are calculated for C by combining the upper bounds of Eq. 9.6 with Eq. 9.7, as shown below:

$$[\mu_C + z.\sigma_C] + v \tag{9.8}$$

Our study detects an attack by comparing incoming traffic with the baseline distribution. Therefore, the modified fast-entropy method signals the presence of

an attack by computing the flooding attack confirmation threshold (Θ) from the normal baseline behaviour of the network. Θ is calculated from the multiple of standard deviation (σ_B) of B with baseline fast-entropy rate (δ_B) using Eq. 9.9.

$$\Theta = \delta_B * \sigma_B \tag{9.9}$$

The selection of appropriate threshold values is essential for the accurate and precise detection of DDoS attacks in the resource-constrained ICPS. The sole purpose is to decrease the rate of false alarms. The selection of threshold values is discussed in Sec. 9.6.3.

9.6 Experimental Analysis and Discussion

9.6.1 Experimental Configurations and Datasets

Our proposed approach was implemented on Raspberry PLC 50RRA with 4GB RAM. The proposed algorithm was codified in Python language and run many times on two different publicly available DDoS datasets: BOUN DDoS 2020 (Erhan & Anarim, 2020), and CICDDoS 2019 (Sharafaldin et al., 2019), which are used to detect different types of DDoS attacks. Note, we will use BOUN DDoS and CICDDoS to represent BOUN DDoS 2020 and CICDDoS 2019 datasets, respectively.

The BOUN DDoS and CICDDoS datasets include the recent DDoS attack vectors or types. Several studies have used them for real-world intrusion detection (Zahid, Kuo & Sinha, 2021a; Ali et al., 2021). The BOUN DDoS dataset was generated to detect flooding DDoS attacks like TCP-SYN and UDP flooding. In contrast, the CICDDoS dataset was used to detect Exploitation and Reflection-based DDoS attacks. For our work, shown in Table. 11.2, we have categorised

Table 9.2: Information about used datasets

Datasets	Attack Frequency	Attack Start-Time	Attack End-time	Destination IP Address
BOUN_1	1000Hz	75.5356	105.685	10.50.199.99
BOUN_2	2000Hz	280.594	303.162	10.50.199.99
BOUN_3	1500Hz	180.942	203.552	10.50.199.99
CICDDoS	Variable	12:30:00	14:35:00	192.168.50.4
CICDDoS	Variable	10:30	13:30	192.168.50.1

the BOUN DDoS dataset as BOUN_1 for attack period 1 (75.535-105.685secs) with attack frequency 1000Hz, BOUN_2 for attack period 3 (280.594-303.162secs) having an attack frequency 2000Hz and BOUN_3 for attack period 2 (180.942-203.552secs) having an attack frequency 1500Hz. Similarly, the CICDDoS dataset contains the SYN flooding attack samples captured at 12:30:00-14:35:00 on 12^{th} January and 10:30-13:30 on 11^{th} March.

Moreover, most of the existing works merged two different datasets to differentiate the legitimate and attack traffic, which influence their detection results (Tsobdjou et al., 2022). However, we have used the same datasets for both normal and attack traffics to analyse our proposed work.

9.6.2 Simulation Scenarios and Results

To show the applicability and flexibility of our proposed technique, we experimented with multi-scale traffic characteristics and considered three scenarios:

Scenario 1- Attack Rates: The packet transmission characteristics determine the attack rates. There are two types of attack rates: Predictable and Non-predictable attack rates. The predictable attack rate involves sending the attack packets to the victim according to a predictable pattern. In contrast, the non-predictable attack rate involves sending the attack packets randomly or at a variable rate to avoid being discovered. In our study, we have used the attack rates of the BOUN DDoS dataset as the predictable DDoS attack rates, and the CICDDoS dataset is used to validate our approach for non-predictable attack

rates. The simulation results of our detection technique for attack rates scenarios are shown in Table. 9.7.

Scenario 2- Attack density: An attack density is a ratio between the number of attack packets compared to the number of normal packets at the time of the attack (Erhan & Anarim, 2020). The performance of the proposed detection technique for different attack densities is shown in Table. 9.7.

Scenario 3- Attack periods: We have performed experiments with different attack periods, such as 10sec, 15sec, 60sec, 120sec. Table. 9.7 depicts the results of different periods for each dataset.

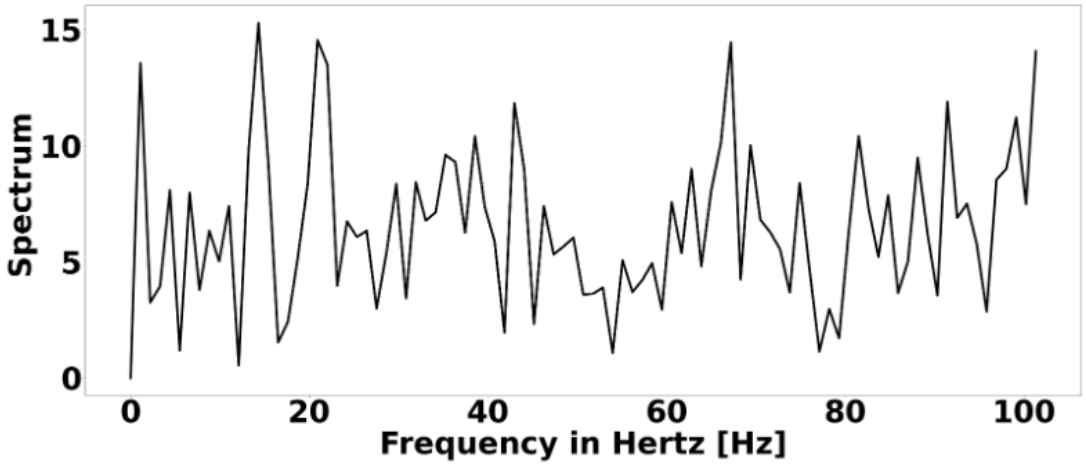
The proposed technique was tested for the mentioned datasets and simulation scenarios. The first step in our experiments is to set the baseline by following the procedure mentioned in (Sec. 9.5.1). For illustration, the normal magnitude spectrum with the highest mean (μ=6.5) was selected as a baseline in the CICDDoS dataset, as shown in Fig. 9.2(a). This baseline is used to compute the flooding attack confirmation threshold ($\Theta = 11.12$).

Table 9.3: Simulation results for unpredictable attacks in ten bins utilising CICDDoS dataset

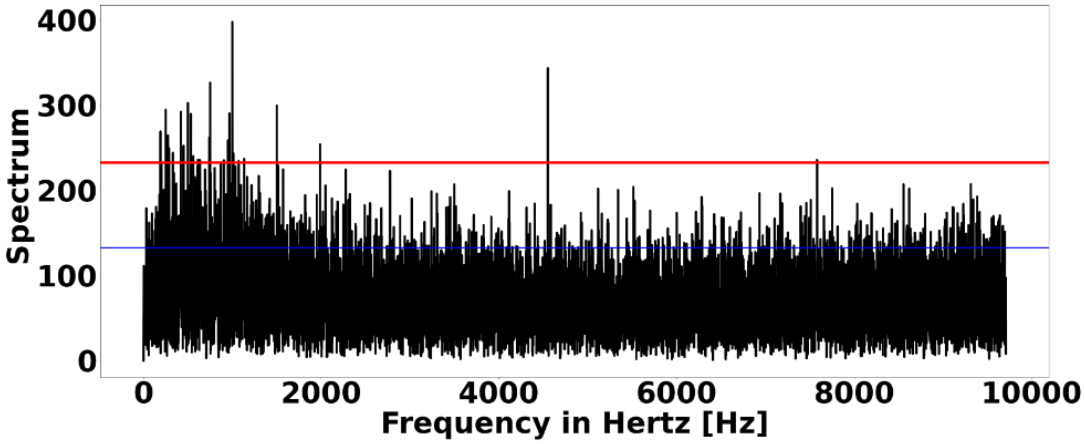
Results	**Bin1**	**Bin2**	**Bin3**	**Bin4**	**Bin5**	**Bin6**	**Bin7**	**Bin8**	**Bin9**	**Bin10**
J	**0.1**	**0.25**	**0.2**	**0.1**	**0.01**	**0.3**	**0.25**	**0.05**	**0.1**	**0.2**
C_{max}	**422.4**	**328.3**	**339.8**	**330**	**353.5**	**364.6**	**439.1**	**403.4**	**355.1**	**397.7**
ζ_Z	**8.95**	**9.35**	**9.31**	**9.34**	**9.25**	**9.16**	**8.85**	**9.01**	**9.20**	**9.04**
δ_Z	**6.99**	**6.96**	**6.9**	**6.96**	**6.93**	**6.97**	**6.89**	**6.99**	**6.93**	**6.96**
Θ	**11.12**	**11.12**	**11.12**	**11.12**	**11.12**	**11.12**	**11.12**	**11.12**	**11.12**	**11.12**
No of dominant peaks	**7**	**12**	**15**	**17**	**12**	**10**	**3**	**5**	**9**	**9**

Using the *CICDDoS* dataset, we conducted a simulation of the flooding attack on a victim's destination IP and used our technique to detect the attack. The purpose was to determine the presence of an attack in the incoming traffic. For example, using the steps discussed in Sec. 9.4.1, we simulate an attack with a period of 60 secs and set the window size (t_w) to 1 sec. During this attack period, 1582112 packets were captured with a 70% attack density, and 60 bins of 1 sec

were observed.



(a) Baseline



(b) Incoming traffic with β (blue line) and ξ (red line)

Figure 9.2: Illustration of Baseline and Incoming Traffic Magnitude Spectra

Table. 9.3 shows the simulation results of 10 bins generated within first 10 secs and Fig. 9.2(b) shows the incoming traffic spectrum for Bin 10. Jaccard similarity between the baseline and incoming traffic is $J = 0.2$, which shows significant variance. Therefore, the system is considered under flooding attack. To further confirm the presence of the flooding attack within the incoming traffic, as shown in Fig. 9.2(b), the modified fast-entropy was calculated for amplitude values greater than or equal to the lower threshold value $\beta = 121.8$, represented

by a blue line in Fig. 9.2(b). It is evident from Table. 9.3 that incoming traffic potentially indicates a flooding attack as δ_Z (6.96) is less than Θ (11.12). After confirming attack presence, attack volume is identified for the values greater than or equal to the upper control threshold value $\xi = 235$, represented by a red line. The dominant peak's fast-entropy value ($\zeta_Z = 9.04$) for the incoming traffic is compared with other fast-entropy values within Bin 10. Finally, an alert will be generated when our proposed approach identifies the number of dominant peaks that are contributing to the intensity of an attack.

Similarly, Table. 9.4 shows the simulation results for BOUN_2 dataset for 10 secs where attack density was 40% and attack rate was 2000 Hz (predictable attacks). We also applied our approach on BOUN_1 and BOUN_3 datasets and complete simulation results are available in (detracted for the blind review)

Table 9.4: Simulation results for predictable attacks in ten bins utilising BOUN_2 dataset

Results	**Bin1**	**Bin2**	**Bin3**	**Bin4**	**Bin5**	**Bin6**	**Bin7**	**Bin8**	**Bin9**	**Bin10**
J	**0**	**0.1**	**0.3**	**0**	**0.2**	**0**	**0.2**	**0.3**	**0.1**	**0.3**
C_{max}	**772.6**	**662.9**	**1115.1**	**724.06**	**900.5**	**703.9**	**793.3**	**799.6**	**886.1**	**865.3**
ζ_Z	**3.17**	**3.5**	**2.5**	**3.34**	**2.8**	**3.35**	**3.04**	**3.11**	**2.90**	**2.95**
δ_Z	**6.0**	**6.4**	**6.16**	**6.24**	**6.39**	**6.23**	**6.03**	**6.25**	**6.11**	**6.22**
Θ	**15.08**	**15.08**	**15.08**	**15.08**	**15.08**	**15.08**	**15.08**	**15.08**	**15.08**	**15.08**
No of dominant peaks	**1**	**2**	**1**	**2**	**1**	**2**	**3**	**1**	**2**	**2**

We also performed the same tests for BOUN_1 and BOUN_3 datasets and complete results are available in (detracted for the blind review).

9.6.3 Selection of Thresholds values

To determine the appropriate values for the lower control threshold (β) and upper control threshold (ξ), the critical selection was the z (integer) range of standard deviation (range of [-6,+6] is commonly used in practice). While determining thresholds, the integer z is denoted as z_β for calculating β and as z_ξ for computing ξ. Tables. 9.5 and 9.6 show simulation results with different values for z_β and z_ξ,

Table 9.5: Determination of Values for CICDDoS dataset

Value of z_β	**Value of z_ξ**	**Avg Accuracy**	**Avg Detection time**
1	2	40%	0.8sec
	3	75%	0.75sec
	4	90%	0.65sec
	5	95.5%	0.6sec
	6	Not Applicable	
2	3	85%	0.7sec
	4	93%	0.63sec
	5	100%	0.6sec
	6	Not Applicable	
3	4	60%	0.7sec

Table 9.6: Determination of Values for BOUN DDoS dataset

Value of z_β	**Value of z_ξ**	**Avg Accuracy**	**Avg Detection time**
1	2	20%	0.45sec
	3	45%	0.47sec
	4	50%	0.43sec
	5	55%	0.39sec
	6	55%	0.44sec
2	3	82%	0.39sec
	4	95%	0.35sec
	5	100%	0.25sec
	6	80%	0.4sec
3	4	70%	0.38sec
	5	75%	0.38sec

and the optimal selection was based on the average accuracy and attack detection time. It is shown in Table. 9.5 that z plays an essential role in the detection of flooding attacks. If we select a lower value for z, the system becomes more sensitive to minor deviation and generates an alert, resulting in less accuracy. Similarly, setting a higher value result in an alert when there is a substantial deviation from the baseline. However, the system might lead to inaccuracy. In our work, the lower threshold is used to remove noise and the upper threshold is used to select the amplitude values representing the attack intensity in terms of dominant peaks. The need is a trade-off between accuracy and detection time taken by the

proposed approach. Therefore, we select 2-sigma (2-standard deviation) for β and 5-sigma (5-standard deviation) for ξ for both datasets, shown in Table. 9.5 and Table. 9.6.

9.6.4 Complexity Analysis

Let n be the total number of packets within an attack period t_w. Our detection approach monitors k samples ($k < n$) in time-interval t_s. In other words, we obtained the detection result within t_s for k samples. Let the runtime of computing these samples be defined as α, i.e, $\alpha = t_s \times k$. The spectrum calculation is based on QuickDFT, which has a time complexity of *nlogn* (Zahid, Kuo & Sinha, 2021a), but in our case (as mentioned above), the time complexity is O(αlogα). The spectrum analysis includes the time complexity of Jaccard similarity method $O(\alpha)$, modified fast-entropy method ($O(\alpha\log\alpha)$), and Euclidean distance calculation $O(\alpha)$. Now by combining all these time complexities we get: $O(\alpha) + O(\alpha) + O(\alpha\log\alpha) + O(\alpha\log\alpha)$ $= 2(O(\alpha)) + 2(O(\alpha\log\alpha)) = O(\alpha\log\alpha)$. This time complexity indicates that our light-weight active security technique is an efficient solution for detecting multi-scale flooding attacks in resource-constrained ICPS.

9.6.5 Performance Evaluation

To assess the performance of the proposed detection technique, we have used four performance evaluation criteria: false positive rate (FPR), true positive rate (TPR), accuracy (Ac) and precision (P). *False positive rate* (FPR) is the percentage of normal instances that are incorrectly classified as an attack and is calculated by Eq. 9.10.

$$FPR = \frac{FP}{TN + FP} \tag{9.10}$$

True positive rate (TPR) is a ratio of appropriately identified attack instances

to the total number of attack instances in the dataset. It is also called sensitivity and is computed by using Eq. 9.11.

$$TPR = \frac{TP}{TP + FN} \tag{9.11}$$

Accuracy (Ac) is a metric used to describe how correctly the technique detects attacks, and it is computed by Eq. 9.12.

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{9.12}$$

Precision is a measure of how well a system can detect attacks or normal behaviour. It is calculated by using Eq. 9.13.

$$P = \frac{TP}{TP + FP} \tag{9.13}$$

In the above-mentioned performance measures, TP means True positive (result indicating correct identification of flooding attack traffic), where FP means false positive (result showing incorrect identification of normal traffic as an attack traffic), where TN means True negative (correct identification of normal traffic), where FN means False negative (incorrect identification of attack traffic as normal traffic).

The results illustrated in Table 9.7 show that the proposed technique has 100% results for the BOUN DDoS and CICDDoS datasets during different attack scenarios.

9.6.6 Comparison with Existing Methods

A general comparison of the proposed technique with the existing literature was conducted in Section 9.3. Since many approaches exist in the literature, conducting a fair comparison with all available works is difficult. Instead, the results of the proposed work are compared with the approaches referred in (Tsobdjou et al., 2022; Zahid, Kuo & Sinha, 2021a; B. Liu et al., 2022; Ali et al., 2021), as illustrated in Table 11.12.

Overall, the results demonstrated that our approach exhibits robust performance, maintaining optimal accuracy under multi-scale flooding attacks without requiring complex or expensive implementations compared to current works. We have implemented our work on PLC and present the performance overhead (memory and CPU usage) and attack detection time, showing that our proposed approach is outperforming, resource-efficient, and flexible to apply in resource-constrained environments. The results also demonstrate that multi-method attack detection proves better than single-method detection. For example, our study found that the Jaccard similarity alone was unreliable for detecting attacks; however, when combined with other methods, we can achieve 100% accuracy. Lastly, unlike existing approaches, considering attack volumes adds a valuable dimension to our proposed approach by providing a better understanding of attack intensity. It enables to take precise and timely/immediate actions against the potential high-rate attack.

9.6.7 Limitation of the Proposed Technique

The primary purpose of the proposed technique is to detect flooding attacks where incoming traffic exhibits a predictable or unpredictable high-rate traffic. These attacks include various attack densities and generate substantial volume of traffic.

Selecting appropriate dynamic threshold values plays a vital role in accurate attack detection. The performance of the proposed work can deteriorate for low-rate attacks based on the selected thresholds when the background baseline traffic is similar to the attack traffic.

Also, an imbalanced dataset with an unequal distribution of benign and traffic samples could impact the detection's decision. In the future, we will address this factor to optimise our proposed approach further.

9.7 Conclusions and Future Work

Resource-constrained ICPS needs light-weight mechanisms to actively (dynamically and programmability) detect the multi-scale flooding attacks and attack volumes early in the incoming traffic flow. In this study, we used statistically robust and low computational overhead methods to detect multi-scale flooding attacks in the resource-constrained ICPS. We have experimented our work on publicly available datasets. Furthermore, identifying attack volumes to determine the attack intensity introduces a significant aspect integral to our proposed approach. The implementation of our proposed technique on PLC-based systems and the experimental results indicate that our proposed technique is light-weight and surpasses current methods' performances by having an attack detection time of less than 1sec.

Considering the limitations of our work, future work is devoted to developing a fast and efficient online learning model to detect and mitigate the slow-rate DDoS attacks in resource-constrained ICPS.

Table 9.7: Results for different attack rates, attack densities and attack periods

Scenarios	**Attack Rates**				**Attack Density**				**Attack Period**			
Datasets	BOUN_1	BOUN_2	BOUN_3	CICDDoS	BOUN_1	BOUN_2	BOUN_3	CICDDoS	BOUN_1	BOUN_2	BOUN_3	CICDDoS
	1000	2000	1500	Variable	95 (28%)	120.31 (40%)	55 (9%)	350.10 (70%)	20	15	20	60
TPR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
FPR	0	0	0	0	0	0	0	0	0	0	0	0
Ac	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
P	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table 9.8: Comparison with the existing works

References	**Domain Analysis**	**Vol**[1]	**A.Ac**[2]**%**	**Dataset**	**Detect**[2] **time**	**Memory usage**	**CPU usage**
(Tsobdjou et al., 2022)	Time	No	100	No	9.7sec	NG	NG
(Zahid, Kuo & Sinha, 2021a)	Freq	No	NG	BOUN DDoS 2020	576sec	NG	NG
(B. Liu et al., 2022)	Time	No	NG	No	28sec	NG	NG
(Ali et al., 2021)	Time	No	98	CICDoS 2019	NG	NG	NG
Our	Freq	Yes	100, 100	CICDDoS 2019, BOUN DDoS 2020	0.6, 0.25	3MB, < 1KB	10%, 4%

[1] Attack Volume, [2] Average Accuracy, [3]Detection time

Chapter 10

Prelude -Manuscript 5

The following chapter has been submitted as a journal paper in the *International Journal of Information Security (Submission ID 1281cb87-9e48-445a-842e-dd6e33b473eb)* under the title of *Light-weight Slow-Rate Attack Detection Framework for Resource-Constrained Industrial Cyber-Physical Systems*. In general, manuscript 5 addresses the fourth and fifth research objectives through designing, developing, and implementing the light-weight active security solution tailored for resource-constrained ICPS (Section 1.2.1).

Slow−Rate Attack (SRA) poses a substantial threat to resource-constrained ICPS, and the emphasis on their detection is still in the early stages. These attacks look like a legitimate user having a slow connection or a device possessing limited transmission capacity. We have performed a literature review of the ICPS-specific and generic studies that have utilised Machine learning (ML)/Deep learning (DL) mechanisms for SRA detection. Our survey research gaps indicate that the research for actively securing resource-constrained ICPS is in its early stages and has neglected significantly the performance and resource utilisation aspects.

One of the significant contributions of the paper is the design, development,

and implementation of an optimised OSELM-based light-weight framework for binary and multi-class SRA detection actively. OSELM is a fast, efficient, and online learning model using single hidden layer feed-forward neural networks. The proposed framework comprises data collection, training, and prediction components. The training component has two sub-components: data pre-processing and SRA detection training. The SRA detection training sub-component utilises a novel proposed stratified-k fold training method, which completes training in fewer iterations and focuses on the performance and accuracy of attack detection. The prediction component includes the data pre-processing, SRA detection evaluation and inference sub-components. The proposed framework is experimented on the publicly available dataset using PLC. The experiment results for both binary and multi-class detection demonstrate optimal accuracy, outperforming performance and minimal attack prediction time compared to the state-of-the-art methods. The space complexity of our optimised model shows that this model is light-weight and well-suited for the detection of slow-rate attacks in resource-constrained ICPS.

Chapter 11

Light-weight Slow-Rate Attack Detection Framework for Resource-Constrained Industrial Cyber-Physical Systems (Manuscript 5)

11.1 Abstract

ICPS are heterogenous computer systems interacting with physical processes in an industrial environment. The high degree of distribution and interconnectedness poses significant security threats to ICPS, including SRA. SRA often exploit system vulnerabilities arising from the inherent limitations of resource-constrained ICPS computers like programmable logic controller (PLC). Existing literature identify several challenges in detecting and classifying SRA in resource-constrained ICPS. In this article, we propose an optimised OSELM-based, novel light-weight

active security framework for SRA detection. We reduce the memory and space footprint of OSELM through optimisation techniques for deployment in resource-constrained ICPS. Also, we introduce a simple stratified k-fold cross training method to improve the binary and multi-class SRA detection capability of the proposed framework, focusing on performance and accuracy of attack detection. Experimental results demonstrate that our proposed framework can effectively and efficiently detect binary and multi-class SRA with an accuracy of 0.975 and 0.96 with average detection times of 0.03 and 0.04 sec, respectively, using less than 3 MB of memory and 10-12% of CPU usage in PLC-based ICPS.

11.2 Introduction

Low-rate Denial of Service attack (LDoS) attacks are recurring internet security problems in which an attacker exploits the vulnerabilities in a protocol or application through low-rate attack traffic that is difficult to differentiate from benign traffic (Zhijun, Wenjing, Liang & Meng, 2020). LDoS attacks differ from traditional DDoS attack in three ways: (1) These attacks have the same characteristics as legitimate network traffic, making them highly concealed and challenging to detect. (2) The average traffic rate is significantly low and the number of sent packets is extremely small; consequently, these attacks are challenging to detect with traditional mechanisms. (3) LDoS attack utilises TCP/IP protocol's three-way handshake mechanism where control measures look normal; however, this can reduce the quality of service (Zhijun et al., 2020; M. Chen, Chen, Wei & Chen, 2021). LDoS attack is categorized into Quality of Services (QoS), SRA, and Service-Queue attacks based on the damage to the protocols (Rios et al., 2022). SRA slow down application layer protocols by exploiting the methods and characteristics of these protocols' architecture and/or by causing buffer overflow

(sharp increase in request-servicing time) (Zahid et al., 2022a).

SRA are a significant threat to resource-constrained systems that operate in industrial environments. These attacks mimic a legitimate user with a slow connection or a device with low data transmission capacity (Tripathi & Hubballi, 2021; Reed et al., 2021). SRA can cause degraded performance, sensors or actuators data manipulation, system instability, and operational disruption (Zahid et al., 2022a). Mitigating these attacks by modifying the operation of a protocol is not feasible as it requires modifications in the corresponding Request For Comments (RFC), which is a cumbersome process requiring extensive deliberations and discussions between stakeholders (Tripathi & Hubballi, 2021).

To detect SRA, various solutions proposed by researchers include ML, DL, Statistical and Cognitional methods (Mittal, Kumar & Behal, 2022). Among them, ML and DL can automatically learn relevant features and identify complex and subtle abnormal patterns from network traffic (Z. Ahmad, Shahid Khan, Wai Shiang, Abdullah & Ahmad, 2021; Saghezchi, Mantas, Violas, de Oliveira Duarte & Rodriguez, 2022; Khraisat, Gondal, Vamplew & Kamruzzaman, 2019).

Our survey (Section 11.3.1) indicates that due to the potential risks posed on critical operations of ICPS by Slow−Rate Attack (SRA), researchers have started focusing on detecting such attacks; however, the research is still in its infancy. Many works use ML and DL models to detect SRA and claim high accuracy (Z. Wang, Li, He & Chan, 2022; Gogoi & Ahmed, 2022; Kemp, Calvert, Khoshgoftaar & Leevy, 2023; Vedula, Lama, Boppana & Trejo, 2021). The existing works are unsuitable for resource-constrained ICPS because they use many features, have multi-layers with many hidden nodes, and/or need backward and forward propagation to set the optimal hyperparameters. These cause high processing and computation overhead beyond the capacity of typical resource-constrained ICPS components. Also, according to our literature survey in Section 11.3.1,

little emphasis has been given to minimising existing models' size and managing the performance overheads in a resource-constrained environment. *This paper aims to propose a novel light-weight active security framework to detect SRA in resource-constrained ICPS* by answering four research questions:

RQ1 Which existing ML/DL based SRA detection mechanisms are suitable for ICPS applications?

RQ2 What are the limitations of current SRA detection mechanisms identified in RQ1 regarding actively securing resource-constrained ICPS applications?

RQ3 How can a light-weight active security mechanism based on ML/DL be provided in resource-constrained ICPS to detect SRA?

RQ4 How can the effectiveness of the proposed light-weight active security mechanism be evaluated and compared to the solutions identified in RQ1?

We answer RQ1 and RQ2 through a literature survey presented in Section 11.3.1. RQ3 and RQ4 are answered (Section 11.4-Section 11.5) through an adapted Design Science research methodology to develop and evaluate the light-weight mechanism to detect SRA in resource-constrained ICPS applications actively (Offermann et al., 2009).

We introduce a framework built upon an optimised OSELM model (Liang, Huang, Saratchandran & Sundararajan, 2006) and a novel and straightforward adapted stratified k-fold cross training method. This framework enhances efficiency and resource-friendliness by reducing the size and complexity of the OSELM model to provide active security to resource-constrained ICPS against SRA. The proposed training method reduces training time and addresses the imbalance dataset issue, enabling the optimised model to excel in online learning efficiently. OSELM is a fast, accurate, and computationally efficient online learning algorithm that can

update and adapt dynamically (G.-B. Huang et al., 2005). The proposed framework pre-processes incoming network traffic, and trains the optimised OSELM model using our proposed training method to learn the binary and multi-class SRA detection. In binary detection, optimised OSELM is trained to categorise the traffic into normal or attack traffic. In multi-class detection, optimised OSELM is trained to classify traffic into normal or different slow-rate attack types: Slowloris, Golden Eye, SlowHTTPTest, and Hulk (Rios et al., 2022). The framework also includes the prediction component where the optimised OSELM is used in the evaluation/prediction of the new and unseen incoming traffic.

PLCs play an important role in executing control software for ICPS and are susceptible to cyber threats, which impact the availability of ICPS. Another major contribution presented in this article is that our proposed framework made the PLC-based ICPS more security-aware. To the best of our knowledge, we have not found any existing literature that offers a solution to actively detect SRA within PLC-based ICPS. This absence is evidence of the improvement in the state-of-the-art.

The primary contributions of this study are:

1. A literature review of existing SRA detection mechanisms in the literature (Section 11.3.1).

2. The design and implementation of a general light-weight active security framework based on an optimised OSELM to detect binary and multi-class SRA on the resource-constrained ICPS (Section 11.4).

3. The performance improvement that resulted in a reduction of the size, resource-utilisation, and attack detection time of the model, enhancing the self-protection abilities of the resource-constrained ICPS.

4. A novel and simple adapted stratified k-fold cross training method with a significant focus on performance and accuracy of attack detection (Section 11.4).

5. An evaluation of the effectiveness of the proposed framework through PLCs and the publicly available CIC-IDS2018 dataset (Sharafaldin, Lashkari & Ghorbani, 2018a) (Section 11.5).

11.3 Literature Review, Selection of Dataset and Model

This section presents the literature review, a rationale, and a description of the selected model used for binary and multi-class SRA detection and the selected dataset.

11.3.1 Literature Review

Table 11.1: Comparative Analysis of recent Slow-Rate attack detection mechanisms. The metrics are Accuracy (Acc), Precision (Pre), and F1-score (F1).

Rf[1]	**Detection Model**	**Algo**[2]	**Approach**	**DF**[3]	**F**[4]	**L**[5]	**H**[6]	**D.T**[7]	**RC**[8]	**Metrics**	**Datasets used**
(Muralee-dharan & Janet, 2021)	GRU	DL	Supervised	srcIP; dstIP; srcport; dst-port; Pro	80	4	320	NG	No	Acc=0.99; Pre=0.99; F1=0.99	CIC-DoS2017
(Fu, Duan, Wang & Li, 2022)	LSTM	DL	Supervised	NG	32	6	Misc	NG	No	Acc=0.944; Pre=0.92; F1=0.94	CIC-DoS2019; UTSA2021
(Asad et al., 2020)	ANN	DL	Supervised	Misc	66	7	Misc	NG	No	Acc=0.98; F1=0.98	CIC-DoS2017
(Hnamte & Hussain, 2023)	CNN; BiLSTM	DL	Supervised	NG	77	3	Misc	1712	No	Acc=0.99	CIC-IDS2018
(Z. Wang, Zeng, Liu & Li, 2021)	DBN-KELM; DBN-EGWO-KELM	SDL[9]	Supervised	packets rate;conn	49	80	110; 100; 251	NG	No	Acc=0.93; Acc=0.97; Pre=0.83; Pre=0.96	CIC-DoS2017
(ElDahshan, AlHabshy & Hameed, 2022)	AoA-ELM; HBA-ELM	SDL	Supervised	Misc	38	1	883; 84; 77;79	NG	No	Acc=0.90; Acc=0.84	CIC-DoS2017

Table 11.1: Comparative Analysis of recent Slow-Rate attack detection mechanisms. The metrics are Accuracy (Acc), Precision (Pre), and F1-score (F1) (continued).

Rf[1]	**Detection Model**	**Algo**[2]	**Approach**	**DF**[3]	**F**[4]	**L**[5]	**H**[6]	**D.T**[7]	**RC**[8]	**Metrics**	**Datasets used**
(Pratomo, Burnap & Theodorakopoulos, 2018)	AutoEncoder	DL	Unsupervised	bytes	NG	1;3;5	200; Misc	NG	No	NG	UNSW-NB15
(Z. Wang et al., 2022)	Triplet CNN	DL	Supervised	NG	NG	Misc	Misc	NG	No	Acc=0.98; Pre=0.99; F1=0.99	CIC-DoS2017
(Gogoi & Ahmed, 2022)	LSTM	DL	Supervised	request; timestamp	NG	3	Misc	NG	No	Acc=0.99	CIC-DoS2017
(Shaik & Kataoka, 2021)	Multi-Autoencoder	DL	Unsupervised	Misc	35	3	NG	200	No	Acc=0.87; Acc=0.95; Acc=0.99; Pre=0.99; F1=0.94	No
(C. Xu, Shen & Du, 2021)	1DCNN; GRU	DL	Supervised	packets rate		10	64	NG	No	Acc=0.97; Acc=0.99; Pre=0.98; Pre=0.97	No

Table 11.1: Comparative Analysis of recent Slow-Rate attack detection mechanisms. The metrics are Accuracy (Acc), Precision (Pre), and F1-score (F1) (continued).

Rf[1]	**Detection Model**	**Algo**[2]	**Approach**	**DF**[3]	**F**[4]	**L**[5]	**H**[6]	**D.T**[7]	**RC**[8]	**Metrics**	**Datasets used**
(Sambangi et al., 2022)	Logistic Regression	ML	Supervised	NG	25			NG	No	Acc=0.90; Pre= 0.90; F1=0.94	CIC-DoS2017; CIC-DoS2019
(Tang, Zhang, Chen & Wang, 2021)	K-Means	ML	Unsupervised	packets rate				NG	No	Acc=0.98	WIDE2018
(Hussain, Saeed, Khan, Aslam & Aljameel, 2022)	K-means	ML	Unsupervised	Misc	NG			NG	No	NG	WIDE2018
(Yan et al., 2019)	LR	ML	Supervised	Misc	NG	1800		NG	No	Acc=0.94	No
(D. Zhang et al., 2019)	PCA; SVM	ML	Supervised	Misc	NG			NG	No	Acc=0.96	No
(Lima Filho, Silveira, de Medeiros Brito Junior, Vargas-Solar & Silveira, 2019)	RF; AdaBoost; DT; LR	ML	Supervised	srcIP; dstIP; srcport; dst-port; prot	26			NG	No	Pre=.99; F1=0.99	CIC-DoS2017; CIC-IDS2018

Table 11.1: Comparative Analysis of recent Slow-Rate attack detection mechanisms. The metrics are Accuracy (Acc), Precision (Pre), and F1-score (F1) (continued).

Rf[1]	**Detection Model**	**Algo**[2]	**Approach**	**DF**[3]	**F**[4]	**L**[5]	**H**[6]	**D.T**[7]	**RC**[8]	**Metrics**	**Datasets used**
(Aamir & Zaidi, 2021)	KNN; SVM; RF	ML	Semi-supervised	traffic rate; delay; CPU utilisation	NG			NG	No	Acc=0.966; Acc=0.92; Acc=0.95	CIC-DoS2017
(Soheily-Khah, Marteau & Béchet, 2018)	K-means; RF	ML	Semi-supervised	srcIP; dstIP	20			1925.5	No	Acc=0.99	ISCX2012
(Jain, Kaur & Saxena, 2022)	K-Means; SVM	ML	Semi-supervised	Misc	11			No	No	Acc=.89; Pre=0.10	CIC-DoS2017
(Kemp, Calvert & Khoshgoftaar, 2020b)	RF;DT; NB;SVM; 5NN;MLP	ML; DL	Semi-supervised	session data	12			NG	No	NG	No
(Kemp, Calvert & Khoshgoftaar, 2020a)	JRP;RF; C4.5D;C4.5N; NB; SVM; 5NN;MLP	ML; DL	Semi-supervised	Misc	12			NG	No	Acc=0.99; Acc=0.94; Acc=0.81	No
(Gunjal, Patel & Ebrahimi, 2022)	MLP; DCN	DL	Semi-supervised	NG	42;36	3	9	NG	No	Acc=0.93; Acc=0.70	CIC-IDS2018; UNBW2015

Table 11.1: Comparative Analysis of recent Slow-Rate attack detection mechanisms. The metrics are Accuracy (Acc), Precision (Pre), and F1-score (F1) (continued).

Rf[1]	**Detection Model**	**Algo**[2]	**Approach**	**DF**[3]	**F**[4]	**L**[5]	**H**[6]	**D.T**[7]	**RC**[8]	**Metrics**	**Datasets used**
(B. Li, Wang, Xu, Cheng & Qin, 2022)	K-means	ML	Semi-supervised	net flow data	NG			NG	No	F1=22.7	CIC-DoS2017; ISCX2012
(Sourbier et al., 2022)	Tangled program graph	ML	Supervised	NA			NG	3847	No	Acc=0.91; Pre= 0.953	CIC-DoS2017; CIC-IDS2018
(Kemp, Calvert & Khoshgoftaar, 2018)	RF;C4.5N; NB;SVM; 5NN;MLP	ML; DL	Semi-supervised	Misc	12			NG	No	Acc=0.96; Acc=0.88; Acc=0.89	No
(Kemp et al., 2023)	C4.5;KNN; NB;RF; JRip		Semi-supervised	net flow data		NG		NG	No	Acc=0.98; Acc=0.96	No
(Vedula et al., 2021)	LSTM; RF	ML; DL	Semi-supervised	interarrival times of net flows	2			NG	Yes	Acc=0.94; F1=0.93	UTSA-2021

Table 11.1: Comparative Analysis of recent Slow-Rate attack detection mechanisms. The metrics are Accuracy (Acc), Precision (Pre), and F1-score (F1) (continued).

Rf[1]	**Detection Model**	**Algo**[2]	**Approach**	**DF**[3]	**F**[4]	**L**[5]	**H**[6]	**D.T**[7]	**RC**[8]	**Metrics**	**Datasets used**
our work	*OSELM*	*SDL*	*Supervised*	*Total forward packet;max inter-arrival time;total bytes used for headers in Fwd direction*	*3*	1	4; 5	*0.04; 0.05*	*Yes*	*Acc=0.975; Acc=0.96; Pre=0.975; Pre=0.97; F1=0.98; F1=0.965*	*CIC-IDS2018*

[1]References, [2]Algorithm, [3]Detection Feature, [4]Number of Features, [5]Number of Hidden Layers,

[6]Number of Hidden Nodes, [7]Detection Time in sec, [8]Resource-Constrained, [9]Shallow Deep Learning

Our survey considered studies including generic mechanisms for detecting SRA attacks, and those that are ICPS-specific using ML/DL. Table 11.1 shows the overall findings of our literature review.

The analysis of our survey indicates that the works based on ML are easy to design and construct, based on one or no layers, and prove to be good at detecting SRA, as shown in Table 11.1. However, the use of these models in the context of resource-constrained ICPS is not well-suited in two ways: first, the effectiveness and availability of ML models compromised significantly in dealing with diverse random traffic while operating in a real-world environment, and second, these models use large number of features to train the models resulting in high computational overheads (W. Wang et al., 2021).

The existing works based on DL models are considered more accurate in detecting SRA because they can automatically extract complex features from incomplete and high dimensional data and maintain the history of patterns, which are helpful in detecting different types of SRA (J. Zhang et al., 2021). However, these mechanisms exhibit slower processing speeds due to forward and backward propagation, involve multiple hidden layers to perform complex mathematical operations, and are sensitive to hyperparameters like epochs, large number of hidden nodes, optimisers, learning rate or stopping criteria. Moreover, these models required a large amount of data for training and generalisation. These factors can lead to the challenges of high computational complexities and memory overheads (Asad et al., 2020; Liang et al., 2006), thus hindering the practical implementation of existing mechanisms in resource-constrained environments.

Several works utilise shallow deep learning (SDL) models for classification because of their proven online learning detection capabilities, less training time and computational efficiency (Z. Wang et al., 2021; ElDahshan et al., 2022; Gunjal et al., 2022). The works (ElDahshan et al., 2022; Z. Wang et al., 2021)

are based on traditional Extreme Learning Machine (ELM). ELM has certain limitations regarding training efficiency, memory requirements, and applicability. It is suitable for generating the best predictor by batch learning, where all the data is available upfront. ELM performs the data training at once and requires all the training samples to be stored in memory. These requirements result in exhaustive computational time, memory overhead and require re-training each time new data arrives. Furthermore, existing ELM-based works are unsuitable for resource-constrained ICPS because of the involvement of many hidden nodes. When the number of hidden nodes increases model gets the overfitting problem, the computational complexity of the model also grows, more memory is required to store the nodes parameter, and lastly, due to batch learning these models do not generalise well on the new test data. In summary, ELM-based works are unsuitable for application in real-time environments for attack detection (Al-Haija, Altamimi & AlWadi, 2024; W. Guo, 2019). Therefore, in this study, we consider adopting OSELM, an extended version of ELM, to meet the continuous or sequential learning needs of real-time applications in ICPS (see Section 11.3.2).

Overall, our survey reveals that the research on detecting SRA in ICPS is still in its infancy, specifically in the domain of PLC-based ICPS. Existing works have prioritised attack detection rates, but pay negligible attention to performance and resource use. The trade-off between accuracy and efficiency for faster and early attack detection is a pressing issue in resource-constrained ICPS. This significantly limits the practical suitability of the existing works within resource-constrained ICPS. Our work aims to overcome these current gaps by providing light-weight framework based on an adapted OSELM model that is further refined for efficiency and resource consumption.

11.3.2 Selection of OSELM for SRA attack detection in resource-constrained ICPS

OSELM is an online extension of ELM that uses single hidden layer feedforward neural networks (G.-B. Huang, Zhu & Siew, 2006). Compared to ELM (discussed in Section 11.3.1), OSELM is specifically designed for streaming or sequential data. OSELM is time and memory efficient as it neither requires re-training upon the arrival of new data nor stores all the training samples in memory (W. Guo, 2019; X. Wang, Tu, Zhao & Shi, 2022). It can process training data in a chunk or one sample at a time. It does not require tuning learning rates, learning epochs, and stopping criteria, making the learning phase significantly faster than traditional neural networks (Liang et al., 2006). Also, literature has shown that OSELM can make accurate predictions (Qaiwmchi, Amintoosi & Mohajerzadeh, 2020; Y. Li, Qiu & Jing, 2018) and is robust to noisy data (W. Guo, 2019; A. R. Lima, Hsieh & Cannon, 2017).

In ICPS, as non-stationary and noisy data is collected, monitored, and processed continuously, a well-trained model cannot remain unchanged, and new data features must be continuously learned (W. Guo, 2019). OSELM's robustness to noisy environments and the ability for online learning from new arrival data depict its adaption to changing industrial processes. Also, because of the limited resources, complicated and time-consuming tasks of model re-training and evaluation can introduce delays due to intensive computational needs and impact timely decision-making (Zahid et al., 2024). OSELM's computational efficiency, small memory footprints, faster and simpler learning capability make it suitable for resource-constrained ICPS applications. Moreover, OSELM can efficiently learn the non-linear relationship between incoming data and their deviations or complex patterns, even if they fall within the linear range, which linear models

miss (Y. Li et al., 2018). SRA manipulate data or systems' parameters and deviate them from expected behavior slowly over time; for example, a slow-rate attack could include the alteration of packet header size, which is not linearly correlated with normal behavior (Further detail in Section. 11.5). Thus, the non-linear relationship modeling and real-time adaptive learning of OSELM could help to reveal the gradually hidden patterns in network traffic, making it easier and an appropriate model for SRA attack detection in resource-constrained ICPS.

The *general architecture of OSELM* has three layers: an input layer, an output layer, and one hidden layer between them. *OSELM deployment* is composed of two phases: boosting and online sequential-learning phases, as shown in Figure 11.1.

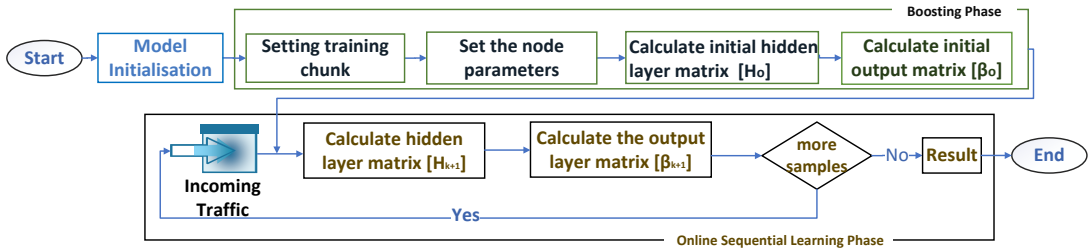


Figure 11.1: Phases of Online Sequential Extreme Learning Machine

In the *boosting phase*, some chunk of training data (equal to or greater than the hidden nodes) is used to train the model using these basic steps (G.-B. Huang et al., 2005): (1) Initialise the parameters of the model randomly, i.e., the random assignment of input weights and biases (2) Calculate hidden layer output matrix (3) Calculate output weight matrix (4) Predict on a new sample.

The *online sequential learning phase* follows the boosting phase where the model learns the train data in chunks or one-by-one by updating the weight matrix incrementally based on the previous weights and new data samples. The details of the basic principles of OSELM can be found in (G.-B. Huang et al., 2005; Liang et al., 2006).

Table 11.2: Information about Benign and Slow−Rate Attacks Types in CIC-IDS2018 dataset

Label	Date	Attack Start-Time	Attack End-Time	Instances
GoldenEye	Thur-15-2-18	9:26	10:09	41,508
Slowloris	Thur-15-2-18	10:59	11:40	10,990
SlowHTTPTest	Fri-16-2-18	10:12	11:08	461,912
Hulk	Fri-16-2-18	13:45	14:19	1,39,890
Benign	15-2-18; 16-2-18			1,442,849

11.3.3 Selected Dataset

In order to select the benchmark dataset for SRA attack detection in ICPS, the dataset must contain real-time network traffic, must be extensive, captures the interaction among various components, has the latest SRA attacks types and considered to be an accurate and effective data for network attack detection (Farhan & Jasim, 2022). Based on these criteria, the CIC-IDS2018 dataset can be used as a benchmark dataset for our work (Sharafaldin et al., 2018a).

The CIC-IDS2018 dataset comprises benign and seven modern-day attacks, including SRA like GoldenEye, Slowloris, SlowHTTPTest, and Hulk. The CICIDS2018 dataset closely mimics the real-world data (PCAPs) gathered by the Canadian Institute for Cybersecurity, having more than 16 million rows and 80 features. The infrastructure for the attack contains 50 machines, whereas the targeted organization comprises five departments, including 420 machines and 30 servers.

This work focused on the network traffic captured on Thursday, February 15, 2018, and Friday, February 16, 2018. The chosen days encompassed different types of slow-rate attacks that were the focus of our study. The information about the labels used to represent slow-rate attack traffic in the dataset is given in Table 11.2. More details of the attack types in each day's dataset can be found in (Sharafaldin et al., 2018a).

11.4 Proposed Light-Weight Slow-Rate Attack Detection Framework

This study proposed a light-weight SRA detection framework, illustrated in Figure 11.2, comprises three major components: data collection, training and

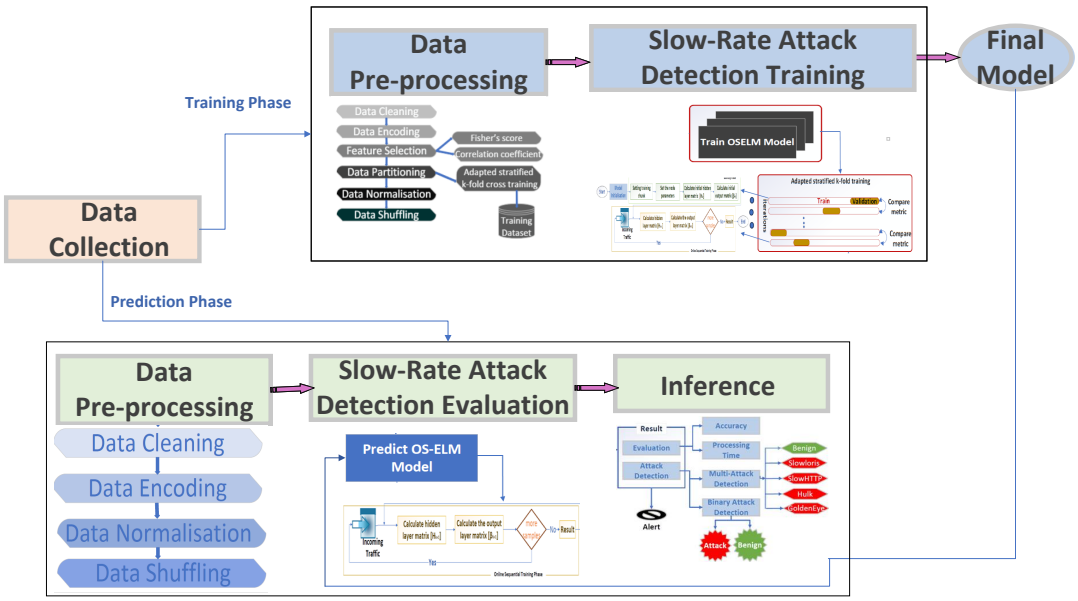


Figure 11.2: Proposed Light-Weight Slow-Rate Attack Detection Framework

prediction. The training component includes data pre-processing and SRA detection training sub-components during the training phase. The predication phase starts once the optimised OSELM model is selected as a final model. In the prediction phase, the prediction component also include a pre-processing sub-component (difference is in Sect. 11.4.2), SRA detection evaluation and inference sub-components.

11.4.1 Data Collection

The first phase is to collect/receive the network traffic from different ICPS sources like sensors and actuators. As a note, the data collection process in a real ICPS

scenario falls outside the scope of this paper. However, in this work, the data collection phase includes selecting the publicly available dataset - CIC-IDS2018 (discussed in Section. 11.3.3). We merged different csv files in CIC-IDS2018 containing benign and SRA network traffic into one file to conduct dataset observation and to proceed to the next phase. The network traffic data contains many features like source and destination IP addresses, ports number, timestamps, and many more. For more information on the features of network traffic in the CIC-IDS2018 dataset, please refer to the (Sharafaldin et al., 2018a).

11.4.2 Data Pre-processing

Data pre-processing is essential to prepare the collected data for later computations. For training purposes, the data pre-processing includes data cleaning, data encoding, feature selection, data partitioning, data normalisation and shuffling (Famili, Shen, Weber & Simoudis, 1997), illustrated in Figure 11.3. During the prediction phase, the data pre-processing sub-component includes data cleaning, data encoding, feature consistency, data normalisation and shuffling, shown in Fig. 11.4.

Data cleaning involves analysing and handling data for missing, undefined, incorrect, or irrelevant information. The data cleaning step is crucial in enhancing the data consistency and quality for efficient processing, providing reliable and accurate results, and reducing biases. For our work, we remove the columns containing only zeros, containing categorical values except for labels, and impute the missing values and infinite values. We retain the duplicate rows from the selected dataset in the data cleaning step. We aimed to ensure effective detection where the model can learn and capture the patterns with similar traits. We made this decision based on the nature of the attack and the patterns that emerge,

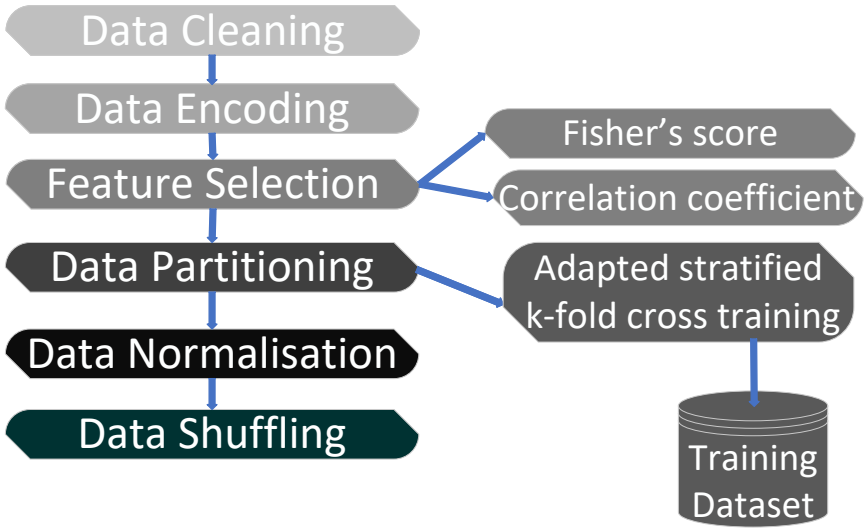


Figure 11.3: Data pre-processing during training phase

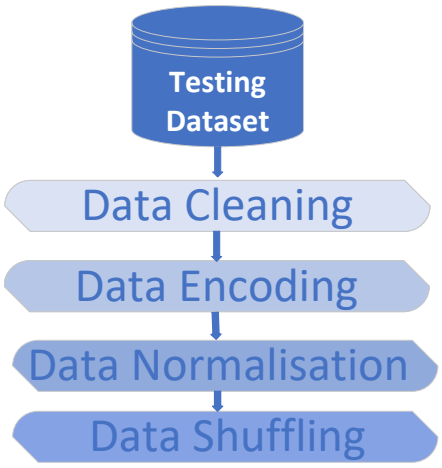


Figure 11.4: Data pre-processing during prediction phase

specifically when an attacker sends multiple packets having similar characteristics or patterns within the same time frame. This consideration is aligned with SRA strategy and result in the effective attack detection.

Data encoding converts categorical features into numerical values because machine learning models only understand the data in integers or float. In our study, we initially applied label encoding on the target label classes (containing the attack names), so the classifier (model) can learn the relationship and patterns

according to their class number. Later, we applied a one-hot encoding scheme that converts multi-class numerical values into binary values (Hnamte & Hussain, 2023). Note, for binary detection, we experiment with and without a one-hot encoding scheme, but finally, we select only a label encoding scheme for binary detection because this scheme is space efficient.

Feature selection is a pre-processing technique to ensure accurate decision making, improve model performance, reduce the prediction time of the model and address the overfitting problem (Z. Wang et al., 2022). It is employed to identify the relevant features used to develop a model that ensures accuracy and generates valuable predictions. In our work, we used a hybrid selection approach. The purpose of the hybrid approach is to determine the relevant sets of features, which are discriminative and possess a strong relationship with the target label classes to understand the patterns for attack detection. First, we used Fisher's score technique (Gu, Li & Han, 2012). This technique uses the Fisher score algorithm to independently determine the subset of best features based on their scores. After that, we applied the Pearson correlation coefficient technique to the subset of features to further select the feature(s) that best correlate with the target (Cohen et al., 2009). This technique assigns a value in a range of $[-1, 1]$, where 0 shows no correlation, 1 shows a perfect positive correlation, and -1 shows a total negative correlation. The purpose of using a hybrid approach is to reduce the likelihood of one technique being sensitive to specific data characteristics; thus, the hybrid approach increases the stability and generalisation of the selected features. The output of this step is used as an input to the attack detection phase.

Data partitioning uses a novel adaptation of stratified k-fold cross-training that addresses challenges arising from having an imbalanced dataset and reduces training time. Imbalanced datasets contain an uneven distribution of target classes, resulting in a biased model with low accuracy due to overfitting (Z. Wang et al.,

2022). To address this issue, we created a *training dataset* (in a `csv` file) containing approximately the same distribution of each target class. Our method splits the training dataset into k folds equally in k iterations. During each iteration, one fold is used as a *validation set*, and the remaining k-1 folds are contained in the *training set*. After each iteration, performance metrics like the accuracy of the validation set and the iteration completion time (training time) are monitored and compared with the previous iterations. Training stops if the accuracy drops or remains the same for consecutive iterations. The final model is evaluated using a separate *testing dataset* with the same distribution of samples. This adapted method often completes training in fewer iterations than the traditional stratified k-cross validation method (Z. Wang et al., 2022) that requires a fixed number of iterations. The rationale for our proposed method is discussed in Sect. 11.5.2. The final model is evaluated using a separate *testing dataset* with unequal distribution of samples.

Data normalisation transforms data values into a common scale (similar range and mean value) to analyse the information easily. We used Min-Max Normalisation (scaling) method, which is a well-known and a simplest method to normalise the range of features by using formulae (Sambangi et al., 2022):

$$x^{'} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{11.1}$$

where x is the scaling data point. In this method, values for each feature are normalised between 0 and 1, where 0 is the minimum value and 1 is the maximum value.

Data shuffling is applied to randomly shuffle the data in the dataset to reduce the bias and improve the model's performance during training and validation.

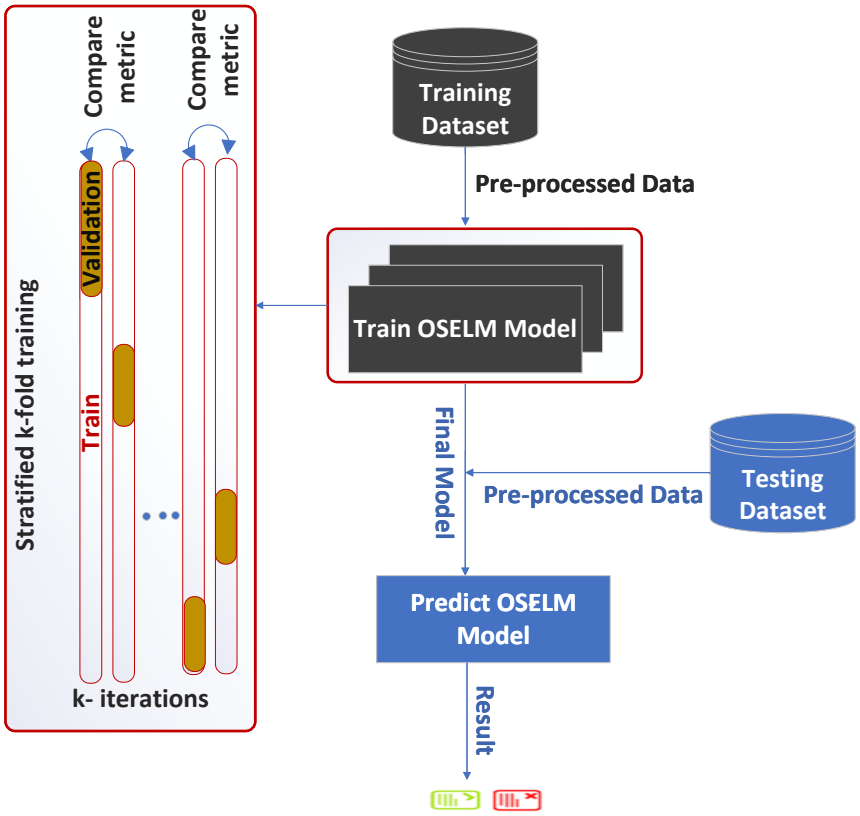


Figure 11.5: Optimised OSELM model training and prediction

11.4.3 Slow-Rate Attack Detection

OSELM model is employed for SRA detection on resource-constrained ICPS. The use of OSELM is two-fold: to detect and classify the network traffic as a benign or slow-rate attack (binary classification) and to classify the traffic based on the slow-rate attack types (multi-class classification). The SRA detection phase has two sub-phases: the training and prediction phases, as shown in Figure 11.5.

Before training the model, we first have to set up the model's architecture with an appropriate number of input nodes, hidden nodes, output nodes, and activation functions. The number of input nodes in the input layer of OSELM is equal to the optimal number of features identified during the feature selection step of the previous phase. The only learning parameter for OSELM is determining

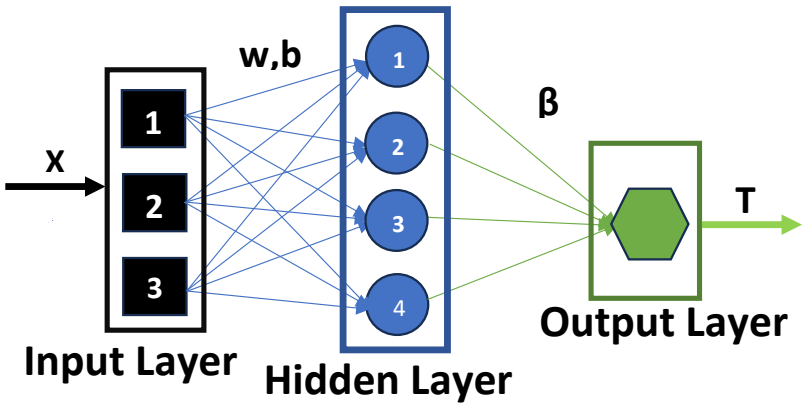


Figure 11.6: Architecture of an Optimised OSELM for Binary Detection with L(hidden nodes), k(input nodes) and o(output nodes)

the optimal number of the hidden nodes. Further detail on the selection of hidden nodes is in Section 9.6. The additive hidden nodes with *sigmoid* activation function are generally used in the hidden layers. The number of output nodes in the output layer is based on the type of attack classification. For binary classification, only one output node and *sigmoid* activation function are used to produce the binary output (benign or attack), as shown in Figure. 11.6. For multi-class classification,

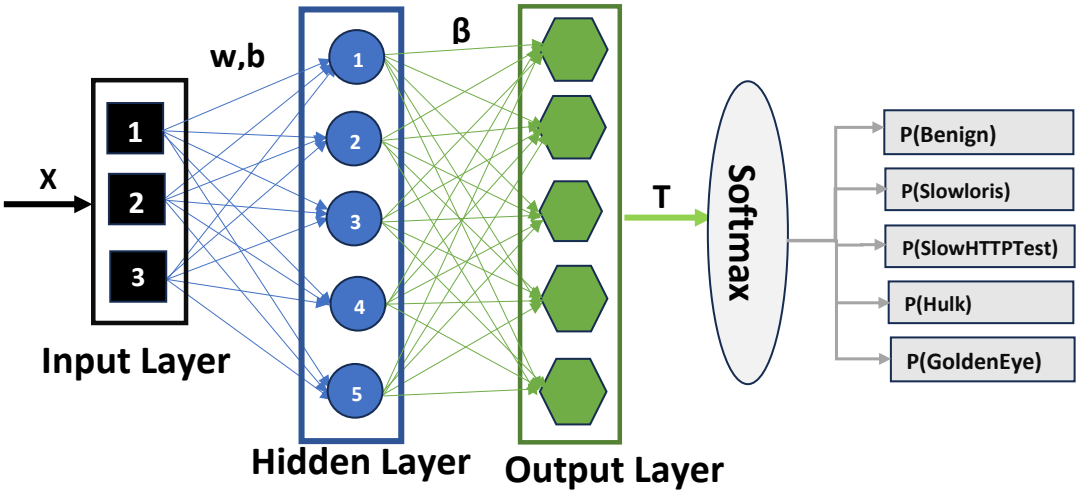


Figure 11.7: Architecture of an Optimised OSELM for Multi-Class Detection with L(hidden nodes), k(input nodes) and o(output nodes), P(probability of target classes)

the *softmax* activation function is used and the number of output nodes is set equal to the number of target classes. In our work, we have five target classes; therefore, we have five nodes in the output layer of our model, shown in Figure 11.7.

Training phase

The training phase involves OSELM training based on adapted stratified k-fold cross training method, shown in Figure 11.5. During each iteration of the training method, the training process starts with the boosting phase (Figure 11.1). In the boosting phase, a chunk of training data is selected for initial learning depending on the optimal number of hidden nodes. First, the weight matrix and biases are initialised randomly to complete the initial training phase. Then, the initial hidden layer output matrix and initial output weight matrix are computed. The details of the basic principles of OSELM can be found in (G.-B. Huang et al., 2005; Liang et al., 2006).

Let N be the training dataset with X distinct samples pairs (x_i, t_i) such that: $N = \{(x_i, t_i)$: $x_i = [x_{i1}, x_{i2} \cdots, x_{ik}]^T \in R^k$ is an input matrix; $t_i = [t_{i1}, t_{i2} \cdots, t_{io}]^T \in R^o$ is a target matrix; $i = 1, 2, \cdots, X$ is a training sample; k is number of input nodes, and o is the number of output nodes$\}$.

Given the chunk of initial training dataset (N_o) with X_o distinct samples pairs selected from N such that: $N_o = \{(x_i, t_i)$: $x_i \in R^k, t_i \in R^o, i = 1, 2, \cdots, X_o; N_o \geq L, L$ is number of hidden nodes$\}$, the steps involved in boosting phase are (G.-B. Huang et al., 2005):

1. First, randomly assign the input weight (w_j) and bias (b_j) for input layer such that $j = 1, 2, \cdots, L$, w_j is a weight matrix $([w_{j1}, w_{j2} \cdots, w_{jk}]^T \in R^k)$ between input and i^{th} hidden nodes and b_j is bias matrix such that $b_j = [b_1, b_2 \cdots, b_k]^T \in R^k$

2. The initial hidden layer output matrix (H_o) is calculated as follows:

$$H_o = \begin{bmatrix} g(x_1.w_1 + b_1) & \cdots & g(x_1.w_L + b_L) \\ \vdots & \vdots \cdots & \vdots \\ g(x_{X_o}.w_1 + b_1) & \cdots & g(x_{X_o}.w_L + b_L) \end{bmatrix} \tag{11.2}$$

When the additive hidden nodes with the activation function $g(x) : R \rightarrow R$ is used, then $G(w, b, x)$ is represented by the following equation:

$$G(w, b, x) = g(x.w + b) \tag{11.3}$$

where g is an activation function and sigmoid function is commonly used as follows:

$$g(x.w + b) = \frac{1}{1 + e^{-(x.w+b)}} \tag{11.4}$$

3. The output weight matrix (β_o) is calculated as follows:

$$\beta_o = M_o H_o^T T_o \tag{11.5}$$

where $M_o = (H_o^T H_o)^{-1}, T_o = [t_1, \cdots, t_L]^T$.

4. Set $k = 0$ where k shows the chunk of training data.

After the boosting phase, the online sequential training phase will start. Upon the arrival of new samples, the hidden layer output matrix and output weight matrix will be updated. This phase will be repeated until all the training samples have been processed. This is to ensure that the model is continuously adapting and learning from the new arrival samples.

Let N_{k+1} is the arriving chunk of data having X_{j+1} number of samples in it such that: $N_{k+1} = \{(x_i, t_i)$: $x_i \in R^k, t_i \in R^o, i = (\sum_{j=0}^{k} X_j) + 1, \cdots, \sum_{j=0}^{k+1} X_j\}$.

Let $y = (\sum_{j=0}^{k} X_j) + 1$ and $z = \sum_{j=0}^{k+1} X_j$.

5. The hidden layer output matrix (H_{k+1}) is computed as:

$$H_{k+1} = \begin{bmatrix} g(x_y.w_1 + b_1) & \cdots & g(x_y.w_L + b_L) \\ \vdots & \vdots \cdots & \vdots \\ g(x_z.w_1 + b_1) & \cdots & g(x_z.w_L + b_L) \end{bmatrix} \tag{11.6}$$

6. Set T_{k+1} by using following equation:

$$T_{k+1} = [t_y, \cdots, t_z]^T \tag{11.7}$$

7. The output weight β_{k+1} is computed as follows:

$$\beta_{k+1} = \beta_k + M_{k+1} H_{k+1} (T_{k+1}^T - H_{k+1}^T \beta_k) \tag{11.8}$$

where $M_{k+1} = M_k - \frac{M_k H_{k+1} H_{k+1}^T M_k}{1 + H_{k+1}^T M_k H_{k+1}}$.

8. Update k=k+1 and repeat from step 5.

After each iteration, the performance of the model is evaluated. When the model achieves the highest accuracy, the model will be saved. Once the model will be trained and saved, prediction phase will start.

Prediction phase

During the prediction phase, the trained model is used to predict data in the testing dataset, which contains new and unseen traffic samples, as shown in Figure 11.8. It is a crucial phase in attack detection as it is responsible for SRA's timely identification and response. Figure 11.8 shows that we first perform the data pre-possessing using the steps discussed in Section 11.4. After pre-processing,

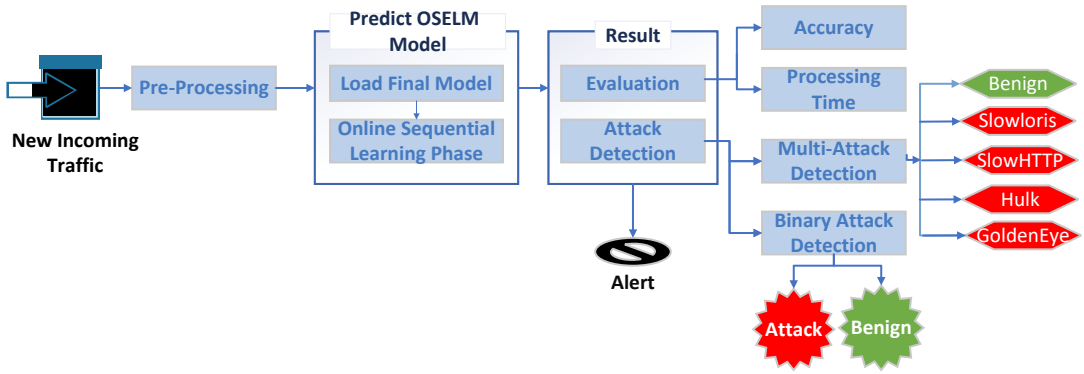


Figure 11.8: Prediction Phase

the final model identified during the previous training phase is loaded and fed with the processed input traffic samples.

During the prediction phase, only online sequential learning phases will be executed. During this phase, the hidden layer output matrix and output matrix will be calculated using Eq. 11.6 and Eq. 11.8, as shown in Figure 11.1. Finally, the model's predicted output (target score) is calculated by Eq. 11.9 (G.-B. Huang et al., 2005).

$$H\beta = T \tag{11.9}$$

where size of β and T is $L \times o$ and $X \times o$ and calculated as:

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_L^T \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1o} \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{L1} & \beta_{L2} & \cdots & \beta_{Lo} \end{bmatrix} \tag{11.10}$$

$$T = \begin{bmatrix} t_1^T \\ t_2^T \\ \vdots \\ t_X^T \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1o} \\ \vdots & \vdots & \cdots & \vdots \\ t_{X1} & t_{X2} & \cdots & t_{Xo} \end{bmatrix} \tag{11.11}$$

The model's output predicts whether the samples belong to benign traffic or attack traffic in the case of binary classification using the Sigmoid function. The Sigmoid function maps the target score to the range $[0, 1]$. If the probability is less than 0.5, the prediction states the value 0 (traffic is benign); otherwise, the traffic is an attack traffic. In the case of multi-class classification, the model classifies the traffic as normal or predicts the attack class type. The predicted target score is mapped using Softmax function. Softmax function converts those target scores into probability distribution $[0, 1]$, where each relative probability shows the belonging of the target score to the corresponding target classes (labels), and finally, the sum of the probabilities is equal to 1. The predicted output with higher probability is considered as an actual output.

Moreover, the result of the model also indicates the accuracy of attack prediction, the processing time needed to detect an attack, and the system overhead to determine the performance of the model discussed in Section 11.5.

11.5 Experimental Analysis and Discussion

11.5.1 Experimental Setup

The experimental setup for training include a Virtual Machine cluster with specification: HP EliteDesk 800 G5 TWR 8591, Kernel Linux 5.19.0-35-generic, Architecture:x86-64, CPU:8, total memory: 31880MB and OS: 64-bit Ubuntu 20.04. The experiments were conducted using Python 3.11 and Visual Studio code. For data-preprocessing, we used Pandas and scikit-learn libraries. For the model implementation, TensorFlow 2.3 and Keras are used in the backend. The evaluation/testing of the proposed framework is done on Raspberry PLC 50RRA with 4GB RAM.

11.5.2 Analysis of Training and Testing Datasets

During the training phase of the proposed SRA attack detection framework, the training dataset is used to train the OSELM model. Based on the stratified k-fold cross training method, the training dataset contains almost equal contributions of each label set. For binary detection, the training dataset has two labels: Benign with value 0 and Attack with value 1; while for multi-class detection, five labels (Benign, SlowHTTPTest, Hulk, GoldenEye, Slowloris) indicate the types of SRA attacks with values as shown in Table 11.4.

Table 11.3: Training and Testing Dataset Information (number of samples, labels and values) for Binary Detection

Label	**Value**	**Training Dataset**	**Testing Dataset**
Benign	0	36,000	1,442,849
Attack	1	36,000	654,300

Table 11.4: Training and Testing Dataset Information (number of samples, labels and values) for Multi-Class Detection

Label	**Value**	**Training Dataset**	**Testing Dataset**
Benign	0	9,000	1,442,849
SlowHTTPTest	1	9,000	461,912
Hulk	2	9,000	1,39,890
GoldenEye	3	9,000	41,508
Slowloris	4	9,000	10,990

The information, like the number of samples and values of each label used in the training dataset for binary and multi-class detection, are shown in Table 11.3 and Table 11.4, respectively. The total number of samples for the training dataset and testing dataset in case of binary detection are 72,000 and 2,097,149 (Table 11.3). For multi-class detection, the training and testing datasets have 45,000 and

2, 097, 149 samples (Table 11.4), respectively. Note, during the data cleaning step, no null or empty data were found. Also, we did not remove the duplicate rows to detect the attack effectively, so the number of samples remains the same (after the data cleaning step) as in the original dataset (discussed in Section11.4).

Moreover, the distribution of samples in the training dataset as training and validation sets are given in Table 11.5.

Table 11.5: Distribution of Training and Validation Set for Binary and Multi-Class Detection

Training Dataset	**Ratio(%)**	**Training set**	**Validation set**
Binary Detection	85 : 15	61, 200	10,800
Multi-Class Detection	85 : 15	38, 250	6, 750

Feature Selection

As discussed in Section 11.4, the selection of the most relevant and informative features is a significant step for accurate detection of SRA, to reduce the computational complexity, and to improve the performance of the proposed SRA detection mechanism.

Table 11.6: Features' ranking based Fisher's algorithm

Feature Name	**Description**	**Rank**
fl_dur	Flow Duration	1
tot_fw_pk	Total forward packets	2
$fw\_pkt\_l\_max$	Maximum size of packet in forward direction	3
fw_iat_max	Maximum time between two packets sent in the forward direction	4
fw_hdr_len	Total bytes used for headers in the forward direction	5
pkt_size_avg	Average size of packet	6
bw_iat_avg	Mean time between two packets sent in the backward direction	7
fw_seg_min	Minimum segment size observed in the forward direction	8
bw_iat_min	Minimum time between two packets sent in the backward direction	9
fw_seg_avg	Average size observed in the forward direction	10

The CIC-IDS2018 dataset has 80 features, and upon performing the Fisher algorithm, ten features were selected based on their scores. The subset of features

corr_matrix	fl_dur	tot_fw_pk	fw_pkt_l_max	fw_iat_max	fw_hdr_len	pkt_size_avg	label
fl_dur	1	0.16	0.11	0.36	-0.14	0.29	0.45
tot_fw_pk	0.16	1	0.11	-0.08	0.97	0.25	0.65
fw_pkt_l_max	0.11	0.11	1	0.11	0.13	0.61	0.26
fw_iat_max	0.36	-0.08	0.11	1	0.7	0.22	0.92
fw_hdr_len	-0.14	0.97	0.13	0.7	1	-0.24	0.78
pkt_size_avg	0.29	0.25	0.61	0.22	-0.24	1	0.37
label	0.45	0.65	0.26	0.92	0.78	0.37	1

Figure 11.9: Features selection based on Pearson Correlation

according to their ranks is shown in Table 11.6. Afterward, to yield more accurate and efficient results, we applied Pearson correlation (Figure 11.9) and identified three optimal features used for binary and multi-class detection. The final features are selected based on a strong positive correlation with labels and a combination of positive and negative correlations with each other. The reason for such a combination is to make the framework robust and effective to capture not only the common and recurring attack patterns but also diverse attack behavior. Figure 11.9 shows that features like total forward packets (fw_iat_max) followed by fw_iat_max and fw_hdr_len have strong correlation with target class label.

The feature, *Total forward packets (tot_fw_pk)*, is a potential indicator to detect SRA in our work. In SRA, an attacker sends a small number of packets over an extended period; thus, an unusually lower rate of forward packets compared to benign traffic indicates abnormalities in the network traffic.

Likewise, if the maximum time interval between two consecutive packets is abnormally high, it indicates a slow traffic rate compared to benign traffic. Therefore, another significant contributor for detecting SRA is the *maximum inter-arrival time between two packets* (fw_iat_max).

In SRA, attackers send low traffic, so the packet header constitutes a significant portion of the overall packet size, resulting in a minimum payload. Our proposed model detects SRA by analysing the *total bytes used for headers in the forward*

direction (fw_hdr_len).

Thus in our work, the final features, from the CICDOS2018 dataset, used to detect SRA are total forward packets, maximum inter-arrival time between two packets, and total bytes used for headers in the forward direction.

Determination of Hidden Nodes

According to the ELM learning theory, the number of hidden nodes (L) should be greater than the feature dimensions (Leng et al., 2015). In our work, with a major focus on resource-constrained ICPS, we determined the optimal number of hidden nodes during stratified k-fold cross training when the validation set gives the maximum accuracy (Section 11.4). With extensive experiments, we observed that with the number of hidden nodes, i.e, $L = 4$, we obtained the optimal accuracy (Table 11.10) for binary detection. In contrast, optimal accuracy is obtained with $L = 5$ for multi-class detection, as shown in Table 11.9.

Initial Training Chunk Size Determination

Determining the initial chunk of data during the boosting phase sets a foundation for the online sequential learning phase and thus enables OSELM to adapt and learn from new data efficiently. According to OSELM learning theory, the chunk size should equal or exceed the number of selected hidden nodes. For our work, the accuracy of the validation set and training time help to determine the chunk size both for binary detection (Table 11.7) and multi-class detection (Table 11.8).

First, for the binary detection, we experimented with keeping the chunk size equal to the number of hidden nodes and then increasing it. Table 11.7 shows the experimental results for choosing different chunk sizes. Finally, when we choose an

Table 11.7: Initial Training Chunk Size Determination for Binary Detection

Initial Chunk Size	Accuracy	Iterations Number
	0.2	Iteration 1
	0.4	Iteration 2
	0.2	Iteration 3
4	0.4	Iteration 4
	0.2	Iteration 5
	0.2	Iteration 6
	0.2	Iteration 7
	0.5	Iteration 1
	0.6	Iteration 2
	0.5	Iteration 3
6	0.7	Iteration 4
	0.7	Iteration 5
	0.7	Iteration 6
	0.79	Iteration 1
	0.98	Iteration 2
	0.98	Iteration 3
8	0.98	Iteration 4
	0.1	Iteration 1
	0.6	Iteration 2
	0.6	Iteration 3
10	0.8	Iteration 4
	0.7	Iteration 5
	0.8	Iteration 6
	0.8	Iteration 7

initial chunk size equal to 8, the performance of the proposed framework reaches close to optimal.

The same pattern we applied for multi-class detection. Our proposed framework produces good results with both initial chunk sizes equal to 7.5 and 10, as shown in Table 11.8. However, due to the resource-constrained nature of ICPS, we opted for the chunk size that involves fewer iterations and training time, i.e, 10. For chunk size 12.5, we got 90% accuracy in the first iteration, but the result fluctuated as

Table 11.8: Initial Training Chunk Size Determination for Multi-Class Detection

Initial Chunk Size	**Accuracy**	**Iterations Number**	**Training time in *sec***
	0.2	Iteration 1	0.6
	0.3	Iteration 2	0.58
	0.35	Iteration 3	0.3
5	0.5	Iteration 4	0.27
	0.6	Iteration 5	0.25
	0.65	Iteration 6	0.25
	0.72	Iteration 7	0.15
	0.72	Iteration 8	0.12
	0.72	Iteration 9	0.11
	0.3	Iteration 1	0.6
	0.49	Iteration 2	0.47
	0.7	Iteration 3	0.23
7.5	0.97	Iteration 4	0.19
	0.97	Iteration 5	0.17
	0.97	Iteration 6	0.15
	0.54	Iteration 1	0.5
	0.8	Iteration 2	0.21
	0.97	Iteration 3	0.13
10	0.97	Iteration 4	0.10
	0.97	Iteration 5	0.10
	0.9	Iteration 1	0.5
	0.7	Iteration 2	0.3
	0.7	Iteration 3	0.25
12.5	0.97	Iteration 4	0.19
	0.8	Iteration 6	0.11
	0.9	Iteration 7	0.10
	0.75	Iteration 8	0.10

we did further iterations.

Selection of the Final Optimised Model for Binary and Multi-Class Detection

Table 11.9 and Table 11.10 show the accuracy, and detection time in sec for the iterations performed during the stratified k-fold cross training. While training to select the final model for the prediction phase, we evaluated the OSELM model performance after each iteration. For example, in the case of binary detection, after the second iteration, the model's accuracy (98%) was compared with the previous iteration (79%). Similarly, we repeat the steps with further iterations. By applying the concept of early stopping, as discussed in Section 11.4, we stopped at iteration 4 and saved the model weights. We can either stop at iteration 3 as

Table 11.9: Multi-Class Detection Results during Training Phase with $L = 5$ and initial chunk size = 10

Phases	**Iterations**	**Accuracy**	**Time in** *sec*
Training	Iteration 1	0.54	0.5
	Iteration 2	0.8	0.21
	Iteration 3	0.97	0.13
	Iteration 4	0.97	0.10
	Iteration 5	0.97	0.10

Table 11.10: Binary detection results during training phase with $L = 4$ and initial chunk size = 8

Phases	**Iterations**	**Accuracy**	**Time in sec**
Training	Iteration 1	0.79	0.6
	Iteration 2	0.98	0.5
	Iteration 3	0.98	0.3
	Iteration 4	0.98	0.25

well, but to ensure the stability of the model, we did one further iteration. The same procedure was applied for multi-class detection and saved the model when the accuracy reached close to the optimal (97%).

Evaluation

Once the best-performing trained model was selected, next the performance of the trained model was evaluated on the testing dataset that contains new and unseen data samples. During the prediction phase, data in the testing dataset follow the

Table 11.11: Performance Evaluation of Proposed Framework

Proposed Framework	**Accuracy**	**Precision**	**F1-score**	**Recall**	**Detection Time (sec)**	**CPU usage**	**Memory Usage**
Binary Detection	0.975	0.975	0.98	0.975	0.03	10 %	2.6 MB
Multi-class Detection	0.96	0.97	0.965	0.97	0.04	12 %	3 MB

data cleaning, encoding, normalisation and shuffling steps of the pre-processing phase discussed in Section 11.4. The prediction phase was conducted on PLC. The result of the trained model was evaluated with performance criteria like accuracy, recall, precision, F1-score, shown in Table 11.11.

We also considered attack average detection time and system overhead (memory and CPU overhead), demonstrating that the proposed model is light-weight and can be easily used for resource-constrained ICPS. Table 11.11 shows the average attack detection time, memory and CPU usage of the PLC when OSELM performs the binary and multi-class detection over the unseen and new data.

Table 11.12: Comparison of proposed binary SRA detection framework with existing approach

Proposed Framework	**Our work**	**(Gunjal et al., 2022)**
Model Used	Optimised OSELM	MLP
Number of Features	3	42
Hidden Layers	1	3
Number of Hidden Nodes	4	9
Balanced Dataset	Yes	No
Accuracy	0.975	0.80
Precision	0.975	0.82
F1-score	0.98	0.45
Recall	0.975	0.45
Detection Time (sec)	0.03	200
CPU usage	10%	NG
Memory Usage	2.6 MB	NG

Motivation for the adapted stratified k-fold cross-training method

The key objective of the adapted stratified k-fold cross-training method is to optimise the selected model's performance in two ways: to reduce training time and address the imbalanced dataset challenge. The rationale for this method

becomes clearer by comparing it with the traditional stratified k-cross validation method (Z. Wang et al., 2022). In contrast to the traditional stratified k-cross

Table 11.13: Comparison of proposed method and traditional method

Folds	**Proposed method**		**Traditional method**	
	Accuracy	**Time**	**Accuracy**	**Time**
1	0.54	0.5	0.35	9
2	0.8	0.21	0.48	6
3	0.97	0.13	0.60	5
4	0.97	0.10	0.73	4.3
5	0.97	0.10	0.80	3
Result	0.97		Average = 0.59	

validation method, the proposed adapted stratified k-fold cross training method provides a dynamic approach to the training process with a variable number of iterations. During each iteration, the training dataset is divided into k-1 folds for the training set, while one fold is used for validation purpose. This method also incorporates an early stopping mechanism to minimise the computational resource usage and prevent overfitting. The training process is stopped if the accuracy remains the same or no longer improves in subsequent iterations. Furthermore, the performance monitoring criteria are also different from traditional method. It is based on comparing the previous iteration's metrics (accuracy and iteration time) rather than averaging the accuracy results from all k iterations. This dynamic approach ensures the continuous improvement of the model's performance.

Moreover, to resolve the imbalanced dataset issues, such as bias or overfitting, the training dataset is created by having an equal distribution of each target class, ensuring that the majority and minority target classes are represented equally. However, unlike the traditional method, the proportion of target classes in each fold is selected randomly rather than the same to reduce the bias.

As an example, Table 11.13 demonstrated the accuracy and training time of our and traditional methods. We obtained an optimal accuracy (97%) at the third iteration. To confirm the result, we did further iterations. Whereas, the

traditional method with a fixed number of iterations $k = 5$ achieved an accuracy of 59% and appeared computationally expensive. Overall, our proposed method brings forth a dynamic, adaptive and performance-driven training process.

11.5.3 Comparison with Existing Literature

The general comparison of our proposed framework has been done with the existing literature in Sect. 11.3.1. Because numerous approaches have been proposed, conducting an impartial comparison among all of them is challenging. This paper compares our proposed work with existing binary and multi-class SRA detection feed forward neural network approaches in the literature (Gunjal et al., 2022; Asad et al., 2020) using different criteria, shown in Table 11.12 and Table 11.14, respectively.

Table 11.14: Comparison of proposed multi-class SRA detection framework with existing approach

Proposed Framework	**Our work**	**(Asad et al., 2020)**
Model Used	Optimised OSELM	ANN
Number of Features	3	66
Hidden Layers	1	7
Number of Hidden Nodes	4	multiple
Balanced Dataset	Yes	No
Accuracy	0.96	0.98
Precision	0.97	NG
F1-score	0.965	0.96
Recall	0.97	NG
Detection Time (sec)	0.04	NG
CPU usage	12%	NG
Memory Usage	3 MB	NG

The Multi-Layer Perceptron (MLP) model, in (Gunjal et al., 2022), is used to detect application layer attacks for secure communication between devices in an

industrial environment and SCADA framework using forty-two features. Similar to the detector utilised in our work, the detector (MLP) is categorised as a shallow feed-forward neural network (Gunjal et al., 2022) and is used for binary detection. However, our work considers both binary and multi-class detection using only three features. Moreover, it is also evident from Table 11.12 that our proposed framework has outperformed by achieving better accuracy (97.5%), precision (97.5%), and F1-score (98%) compared to their accuracy (80%), precision (82%) and F1-score (45%) for binary detection. Our work also demonstrates higher efficiency like binary attack detection is completed in 0.03 *secs*, while their attack detection time is considerably longer at 200 *secs*.

The authors proposed feed forward neural network architecture with seven hidden layers and involved forward and backward propagation to detect multi-class attacks using 66 features (Asad et al., 2020). Our work also deals with multi-class detection, which is more complicated than binary classification (Hasan, Islam, Zarif & Hashem, 2019). However, in contrast to the approach (Asad et al., 2020), our proposed framework performs multi-class detection through forward propagation utilising a single hidden layer and three features only. Table 11.12 shows that our framework's accuracy (96%) is lower than (Asad et al., 2020), i.e., 98%; nonetheless, our work has demonstrated that the trade-off between the performance and accuracy of attack detection can resolve the challenge of utilising the deep learning models on resource-constrained ICPS.

In contrast to our work, the existing works had not shown the memory and CPU overhead of their proposed approaches. They have used many features and did not consider the imbalance dataset issue, which can affect the accuracy of attack detection and result in an overfitting problem. Comparatively, we have evaluated our work on PLC and proved that a reduction in the model size and selection of proper features can improve the performance of systems. Thus, our

light-weight proposed framework is efficient and applicable to detect SRA attacks on resource-constrained ICPS.

11.5.4 Space Complexity of Optimised OSELM

In this section, we have presented the space complexity of our optimised OSELM to show that it can run efficiently in the resource-constrained environment. The space complexity of an optimised OSELM depends upon the number of weights and biases in the network. For each hidden node, we have two parameters: weights (connecting each node in the input layer to each hidden node in the hidden layer), and biases (associated with each hidden node).

For binary SRA detection, shown in Fig. 11.6, we have three input nodes, four hidden nodes, and one output node. We have four weights for four hidden nodes, each connected to three input nodes. This results in 12 weights, i.e., `3 (input nodes) * 4 (hidden nodes)`. In addition to this, for each hidden node, there is one bias, resulting in four biases. Therefore, for four hidden nodes, we have four biases only. The output layers do not contribute significantly to the space complexity. Thus, there would be a need to store only 12 weights and 4 biases, which is generally considered as an efficient use of space.

Similarly, in the case of multi-class SRA detection (Fig. 11.7), there are three input nodes, five hidden nodes, and five output nodes. Here, the total weight would be `3 (input nodes) * 5 (hidden nodes) = 15` and biases would be 5. Therefore, the space complexity of optimised OSELM for multi-class detection is 20 only: `Space complexity = Total weights + Total biases = 15 + 5 = 20.`

Overall, the space complexity indicates that our optimised model is light-weight and well-suited for resource-constrained ICPS in the detection of slow-rate attacks.

11.5.5 Threats to Validity

One of the inherent limitations of OSELM is a generalisation issue, which arises due to the instability of matrix inversions. Consequently, our framework is also subject to this limitation and prompts us to work on the self-improvement of our framework to incorporate or mitigate this issue. Additionally, our focus was to make PLC-based ICPS security-aware. We experimented with minimum features and, based on our promising experimental results, selected only three features and received good results. We can further incorporate more negatively correlated features with existing ones, providing us with broader coverage of unknown attacks. There is a possibility of changes in data distribution or the availability of new features, so it is necessary to perform the feature selection over time to adapt the model. An unreliable performance assessment can occur if there is any difference in the feature composition between training and testing datasets. We can consider this aspect in the future.

11.6 Conclusions and Future Directions

This paper aims to improve the performance of intrusion detection systems by developing an efficient and accurate approach, making it particularly suitable for the dynamic nature of ICPS, where traffic streams are in real-time, and resources are computationally limited. This work proposes a light-weight active security framework to address the challenges SRA poses early in the incoming traffic by leveraging the capabilities of the simple, fast, and accurate online learning algorithm- OSELM, with a particular focus on PLC-based systems. The proposed framework has been experimented on the publicly available dataset. The experiment results for binary and multi-class detection show notable accuracy, low prediction time, and outperform performance when compared to the existing

state-of-the-art mechanism.

In the future, we intend to provide a deep learning-based solution for SRA attack mitigation, which would be capable of meeting the requirements of resource-constrained ICPS. As resource-constrained applications have limited energy so we can further optimise our work to actively secure resource-constrained with minimum computational power.

Chapter 12

Discussion, Future Directions and Conclusions

This chapter presents a comprehensive overview of the contributions made throughout the thesis and discusses the overall contribution of this thesis, which is presented as a **LASD!** (**LASD!**) for resource-constrained ICPS (Section 12.1). The significance of the research (Section. 12.1.2) and various factors that pose threats to the validity of the research findings (Section. 12.1.3) are also included in this chapter. Further, future directions are presented in Section. 12.2 and concluding remarks appear in Section. 12.3.

ICPS operate within critical industrial environments, often with limited resources. Resource-constrained ICPS are susceptible to various cyber attacks, like DoS/DDoS, because of the increasing digitisation, interconnectivity among heterogeneous components, and the lack of inherent security measures. DoS/DDoS attacks are evolving and threatening concerns that directly affect the availability of these systems (Zahid, Kuo & Sinha, 2021a). The timely detection of these attacks is of paramount importance. Implementing general resource-intensive security measures to protect resource-constrained ICPS against DoS/DDoS may

impact the performance of such systems. The performance optimisation and secure operations of resource-constrained ICPS are crucial and challenging. The existing literature has overlooked this particular facet, which serves as our research's motivation.

In this thesis, we followed a multi-phased design science research methodology to achieve our research objectives iteratively (Section. 1.2). This research was initiated with a Systematic Mapping Study (SMS) (Chapter. 2) and a Systematic Literature Review (SLR) (Chapter. 4) to identify the research gaps in providing active security to the resource-constrained ICPS. Generally, the emphasis on providing active security in resource-constrained ICPS against DoS/DDoS attacks is still in its early stages. Consequently, this thesis has made the following notable contribution in this area.

1. We proposed a novel light-weight active security approach (Chapter 4) for monitoring and detecting the presence of DoS/DDoS attacks on resource-constrained ICPS applications by analysis of the frequency characteristics of incoming network traffic (packets), shown in Fig. 5.1 (Chapter. 4). The approach is capable of detecting the attacks without burdening the computational resources, making it resource-efficient. However, the attack detection time was high and the comprehensive classification criteria to identify the cross-layers DoS/DDoS attacks was needed to be included. These limitations were addressed in the next phases of this research.

2. A classification mechanism regarding cross-domain DoS/DDoS attacks in ICPS has been introduced in (Chapter 8). The proposed taxonomy (Fig. 7.1) and attack scenarios could help cyber security stakeholders to comprehend the characteristics of attacks on smart manufacturing systems and design suitable defence strategies. Nonetheless, the taxonomy needed experimental

validation(s) by developing attack detection scheme(s) to detect multi-vector cross-layer attacks. Hence, we developed light-weight active security solutions for detecting flooding and slow-rate attacks on resource-constrained ICPS in the subsequent phases.

3. We proposed light-weight flooding attacks and attack volume detection technique for resource-constrained ICPS (Chapter 10). A two-phase technique has been devised to analyse the presence and volume of DDoS attacks. The multi-method is employed to dynamically identify the attack presence, ensuring robust and accurate attack detection. The proposed approach worked well with multi-scale flooding attacks. However, the proposed work's performance could deteriorate in slow-rate attacks where attack traffic mimics legitimate traffic. In the following phase, we overcome this limitation.

4. Chapter 10 presents the light-weight slow-rate attack detection framework for resource-constrained ICPS. The foundation of the proposed framework is an optimised OSELM model and an adapted stratified k-fold training method. The framework can effectively and efficiently detect binary and multi-class SRA attacks. The dynamic nature of real-world traffic can introduce changes in traffic characteristics. Moreover, attack vectors are evolving continuously. Therefore, over time, it becomes necessary to re-perform the feature selection process so that the model can adapt to the changes. This work can be enhanced by adding or updating negative or positive correlated features, particularly to address zero-day attacks.

12.1 Light-weight Active Security Detector for Resource-Constrained ICPS

12.1.1 Overall Contribution

The overall contribution of this thesis can be represented as a **LASD!** (**LASD!**) for resource-constrained ICPS. The LISM refers to a component, approach, or system incorporating one or more light-weight solutions (such as the solutions we proposed in this thesis) to provide active security to the resource-constrained ICPS. Figure. 12.1 shows the system model (borrowed from IEC 61499 standard

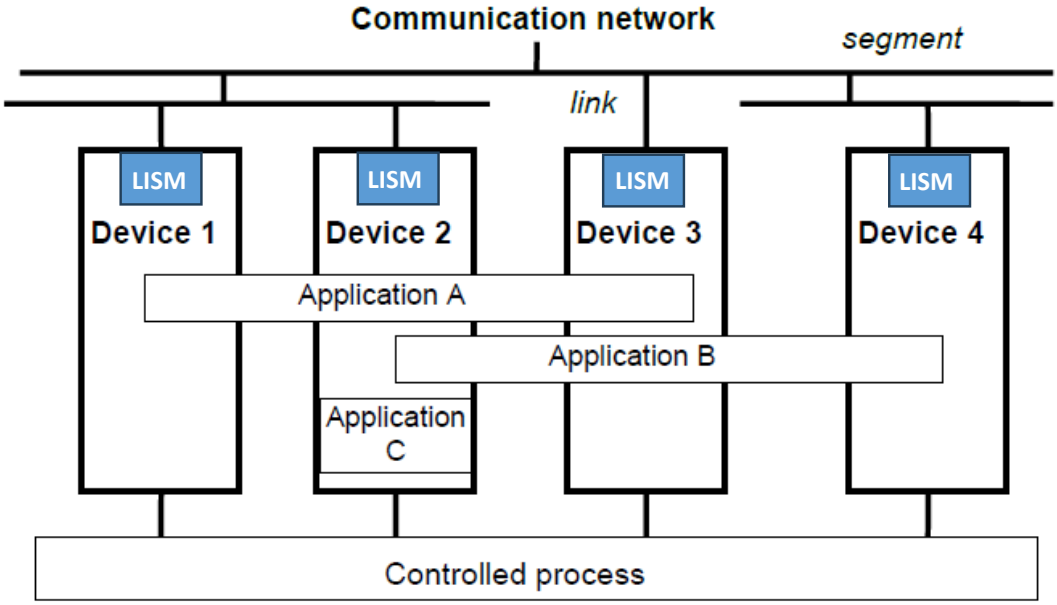


Figure 12.1: *Light-weight Integrated Security Module* (LISM) for Resource-Constrained ICPS

(EN 61499-1:2013 E, 2013)) commonly observed in typical ICPS deployments. In the system model, the blue box is the incorporation of LISM for enhancing the self-protection of each resource-constrained device in ICPS. The system model

comprises several devices (sensors, actuators, controllers) that communicate with each other using network segments and links (communication networks). The three-layered architecture for ICPS consists of a physical, communication, and application layer (Agrawal & Kumar, 2022). The physical layer incorporates the sensors and actuators devices and serves the purpose of timely monitoring and triggering responsive actions. The communication layer integrates various communication networks that support the interconnection of physical and computing devices in an industrial environment. The controllers in the cyber layer are involved in processing information received from the physical layer and generate the decisions to perform specific tasks. Our proposed taxonomy in Chapter. 6 (Zahid et al., 2022a) identifies that the extensive connectivity between ICPS layers leads to various cyber-to-physical, physical-to-cyber, cyber-to-cyber, and physical-to-physical endpoint DoS/DDoS attacks that impact the devices' availability. Similarly, network DoS/DDoS attacks interrupt or degrade devices' communication or impact network bandwidth. Due to the constrained nature of these devices and the lack of inherent security measures, ensuring the security of ICPS, specifically resource-constrained devices, is crucial and challenging. Therefore, instead of generic and complex security measures, efficient and light-weight solutions are required to provide active security by considering resource-limited capabilities. LISM is designed to minimise resource utilisation while providing effective attack detection.

DoS/DDoS attack could be launched by various attack vectors (Zahid et al., 2022a); LISM can deploy multiple algorithms capable of detecting flooding and slow-rate attacks. The timely detection of flooding attacks is essential due to their high operational destructive consequences on the devices and their services. Similarly, slow-rate attacks cause substantial damage at a slow pace, making them challenging to be noticed. With the implementation of LISM, the devices

at each layer of ICPS become able to proactively protect themselves without relying on some external security measures. For example, deploying our proposed LISM within the controllers provides active security to the controller to ensure the reliable operations of the industrial processes. LISM monitors the network traffic, sensor, actuators' data, internal states, and controllers' behaviours. The LISM executes algorithms depending upon the types of attack to be detected. When the detector detects any suspicious activities, it generates alerts to specify the type of attacks.

However, LISM to resource-constrained ICPS presents two essential considerations:

1. Preserving/ensuring the light-weight characteristics of LISM while providing effective and timely detection for both flooding and slow-rate attacks.

2. Identifying the personnel (stakeholders) who will get benefit from the implementation of the LISM.

To ensure LISM can efficiently and effectively detect both flooding and slow-rate attacks while preserving its light-weight characteristics can be viewed from the mode of computations we identified in our conceptual model as discussed in Chapter. 2 (Zahid, Tanveer et al., 2021). The mode of computation specifies the sequence of the executions of the flooding and slow-rate attack detection algorithms within LISM. One way is leveraging concurrent processing capability using multi-threading/multitasking within LISM (Roka & Naik, 2017). Other ways could be sequential and asynchronous computations (L. Liu & Li, 2010). Using a concurrent processing technique within LISM for flooding attacks and slow-rate attack detection algorithms can significantly reduce the time required to detect any malicious behaviour in incoming traffic (Munz & Carle, 2007). Applying the concurrent processing technique, LISM can focus on the detection of both attacks

at the same time rather than sequentially. Moreover, the simultaneous processing of both types of attacks enables the comprehensive monitoring of incoming traffic by ensuring that no attack goes unnoticed. Last, this mode ensures efficient resource utilisation and does not necessarily increase space complexity (Połap, Woźniak, Wei & Damaševičius, 2018). Sequential computation of the flooding attack algorithm followed by the slow-rate attack algorithm is simple and has low computational and space overhead compared to concurrent computation. However, it could introduce a slower response time in attack detection. Using asynchronous computation mode allows the LISM to execute both algorithms concurrently without waiting for the results of each other. This mode efficiently balances resource usage and timely response to attack detection. In general, implementing LISM with any of the above-mentioned computation methods, specifically concurrent and asynchronous modes, could be a strong choice for early attack detection in ICPS by maintaining light-weight characteristics. However, the decision of LISM deployment depends upon the critical infrastructure's architecture, configuration, and operations.

The implementation of LISM, in the context of ICPS, could be beneficial for many stakeholders, including cybersecurity analysts, system security administrators, network administrators, incident response teams, security vendors, solution providers and research and development teams (Faisal, Habib, Hossain, Rashid & Nandi, 2021; Gunduz & Das, 2020; Karter & Hom, 2017; Simplilearn, 2023).

1. System security administrators manage the ICPS and are responsible for the smooth functioning of workplace technology by installing cybersecurity software to secure the devices (kingsland, n.d.; Simplilearn, 2023). They also make recommendations to update the business policies to reduce the impact of cyber threats that could become cyber incidents later. System security administrator could implement LISM to keep the network, devices,

and data secure and available. The alerts generated by LISM indicate the types of DDoS attacks detected and specify the device that gets attacked. This information helps security administrators to apply security controls in a timely fashion and ensure the availability and integrity of ICPS. For instance, security administrators can apply administrative corrective controls by developing and documenting a comprehensive DDoS response plan. They can suggest updating the security policies and procedures and focusing on employee training to take action during an attack. Also, they can apply preventive controls by implementing rate-limiting mitigation measures to limit the requests from a suspicious destination IP.

2. Network administrators are accountable for securing the communication infrastructure of ICPS. They can use LISM for detecting malicious network traffic, including flooding and slow-rate attacks. The attack information aids them to optimise network resources and mitigate security threats. For example, network security is considered from three perspectives: physical security network, technical network security, and administrative network security (Marin, 2005; kingsland, n.d.). Physical network security blocks the hardware that prevents unauthorised network access. In the case of physical-physical and physical-cyber DDoS attacks, identified in the proposed taxonomy (Figure. 7.1), the network administrator can block the physical access point of the malicious sensor, PLC, or actuator to ensure that the adversary will not be able to launch an attack using compromised ICPS devices. Similarly, to provide technical network security, network administrators can divide the network into segments to reduce the impact of the attacks identified by LISM. Also, network administrators can create incident runbooks and playbooks for incident response planning (Onwubiko, 2020).

They can also perform network forensic analysis (Moustafa & Slay, 2018) and apply patch management to prevent attacks using known vulnerabilities.

3. Cybersecurity analysts are responsible for monitoring and securing the ICPS. LISM could provide early attack alerts and help cybersecurity analysts develop protective measures to enhance ICPS security. Cybersecurity analysts could use the information from the alerts and apply traffic filtering rules to block the suspicious IP addresses. They can also enforce security policies that describe the proper use of networks and establish and regularly update incident response plan.

4. Security vendors and solution providers could use LISM as a crucial component to protect the infrastructure offerings to customers and clients. Security vendors can integrate LISM into their security solutions (Emad, 2023). Due to a generic active security solution, they can customise it with additional algorithms according to the needs and requirements of their customers. Upon detection of the attacks, attack details could allow the vendors and providers to create an appropriate incident response plans. They can also generate the attack report and provide it to the client, which could be helpful to recommend countermeasures to them. Thus, LISM could offer them an opportunity to enhance their business by building customer and client trust.

5. Researchers and development teams working on providing active security solutions to ICPS. They can develop their foundations on our light-weight algorithms and models and can further contribute to the development of solutions according to their requirements. Further detail is in the following section (Section. 12.1.2).

In general, the personnel with different roles and responsibilities will benefit from

implementing or using the LISM.

12.1.2 Significance of this Research

This research illustrates the critical importance of active security solutions in resource-constrained ICPS. Integrating resource-constrained cyber-physical components in industrial systems has significantly improved automation and efficiency. The escalation of cyber threats with attacks, including DDoS necessitates providing proactive security measures to protect these resource-constrained ICPS, making it imperative for industry and academic researcher to advance their practices and knowledge to improve society.

Importance of Active Security in Resource-Constrained ICPS

1. **Maintaining Operational Continuity:** The unavailability of resource-constrained ICPS critical operations due to DDoS attacks results in a halt of the industrial processes, productions, delays, and threats to critical infrastructure (Agrawal & Kumar, 2022). Implementing active security solutions (Chapter. 4) ensures the continuity of the services of such systems by detecting and preventing cyber attacks proactively (Zahid, Kuo & Sinha, 2021a). Preserving operational continuity is substantial in resource-constrained ICPS for economic stability, business continuity, and public safety. Integrating LISM in resource-constrained devices is vital in maintaining uninterrupted operational continuity. By monitoring and timely detecting various types of DDoS attacks, LISM enables the cybersecurity stakeholders to take prompt mitigating measures.

2. **Compatibility with Resource Constraint:** Resource-constrained devices need active security solutions compatible with their resource limitations to

combat DDoS attacks (J.-P. A. Yaacoub et al., 2020). Due to the inherent resource-constrained nature of ICPS, protecting them using traditional security measures is challenging (Zahid, Kuo & Sinha, 2021a). Despite the critical component of ICPS, tailoring resource-efficient security solutions for them has not received significant research attention (Xiao et al., 2017; Verma et al., 2023). Research in this thesis mainly focuses on designing and developing light-weight active security solutions that can effectively secure resource-constrained ICPS without imposing a burden on their resources (Chapters. 4, 8, 10).

3. **Latency Reduction**: Implementing LISM in ICPS devices enhances the self-protection of these devices so that they can dynamically and swiftly detect the presence of DDoS attacks and remain accessible to other legitimate devices. The self-protection of the devices reduces the latency, as the data/control information is not sent to the centralised security measure for analysis and attack detection (Q. Chen, Abdelwahed & Erradi, 2013; Yuan & Malek, 2012). This latency reduction is crucial for real-time operations of resource-constrained ICPS without impacting their performance, thus resulting in improved system responsiveness.

Impacts on Industrial and Research Practice

1. **Development of Specialised Algorithm:** Moreover, LISM is a generic security solution rather than a domain-specific one. The versatility of LISM makes it able to provide active security across various domains and in different infrastructures. If the size of ICPS evolves, additional devices can be equipped with LISM to maintain their security coverage. Also, LISM can be enhanced to handle additional algorithms related to various

DoS/DDoS attacks identified in our proposed taxonomy, as discussed in Chapter. 6 (Zahid et al., 2022a). Thus, LISM can be scaled up as needed to accommodate the size and complexity of the ICPS. This scalability is valuable in cybersecurity as new threats are emerging, and LISM can incorporate new algorithms to stay updated with evolving attacks.

2. **Industrial Implications:** Adopting LISM holds influential implications for industry practices (Chapter. 2). For example, the healthcare sector requires uninterrupted access to the patient's information and ensures timely and efficient patient care. Medical devices like wearable medical sensors and actuators are resource-constrained (Rajagopalan, Jagga, Kumari & Ali, 2017). These wearable technologies need light-weight security measures to understand which type of DDoS attack is occurring for their proper mitigation (J.-P. Yaacoub et al., 2020). Implementing LISM in healthcare can provide a defence against DDoS attacks, ensuring the availability of essential medical devices. Similarly, DDoS attacks are also prevalent in the transportation sector (Pretorius & van Niekerk, 2020). For instance, the devices used for controlling traffic signals are also resource-constrained. DDoS attacks on these devices make them fail to manage the traffic flow efficiently and seriously impact public safety and traffic management. LISM can enhance the security in the transportation environment by monitoring real-time traffic data and detecting attacks on time. Thus, our research meets the industry's needs in providing solutions that work within the industrial operational constraint. Beside this, our research could directly benefit industrial practitioners, including cybersecurity personnel, and decision makers, who are responsible for providing the security to organisations, businesses, clients and customers. Different industrial personnel involved

in operations, maintenance, service providers, and cybersecurity analysts could employ LISM to smooth ICPS components' operations as discussed in previous section (Section. 12.1.1).

3. **Innovations by Academic Researchers:** Our research contributions could help the academic community by advancing their knowledge in ICPS cybersecurity. They can further build upon their innovative solutions on our foundations. For example, we have utilised statistical approaches for attack presence and attack volume detection (Chapter. 8). Researchers can employ AI-driven solutions to provide light-weight active security to detect the attacks and their volumes in resource-constrained ICPS. In addition, they have the opportunity to evaluate the performance of our proposed solution with their work, indicating potential improvements. They can also extend or refine taxonomy (Chapter. 6) to identify additional categories of DoS/DDoS attacks targeting each layer of ICPS. Moreover, to provide a holistic approach in cybersecurity for defending resource-constrained ICPS, the researcher can extend our solutions to mitigate the identified attacks. These aspects show that the artifacts developed at the end of each iteration of the design science research methodology (Chapter. 1) could be adopted or extended effectively by academic professionals.

Key Trends in Cybersecurity

1. **Rising Sophistication of DDoS:** DDoS attacks have evolved in sophistication, frequency, types, and sizes (Omer & Pacheco, 2023). The attackers use advance techniques and tactics to invade the detection of these sophisticated attacks. To efficiently and effectively manage infrastructures to combat DDoS attacks, integrated active security solutions are required that can offer

protection against different types of DDoS attacks. LISM is a cost-effective solution enabling critical sectors to defend themselves against multi-vector DDoS attacks. In this way, our research addresses a critical and current issue within cybersecurity discipline.

2. **Adopting AI-driven and Statistical Approaches:** In this thesis, the online learning capability of optimised OSELM and dynamic attack detection capability of employed statistical approaches enable LISM to continuously learn and enhance their detection abilities for new and previously unknown attacks (Chapter. 10). AI-driven DDoS attack detection has proved effective and stands at the forefront (Sarker, Furhad & Nowrozy, 2021; Ansari, Dash, Sharma & Yathiraju, 2022). At the same time, statistical approaches are well-known for analysing the trend of malicious activities in network traffic (Sanna Passino, Adams, Cohen, Evangelou & Heard, 2023). Thus, by harnessing the strengths of AI-driven and statistical-based solutions, LISM ensures that the DDoS attacks are addressed promptly and effectively. Thus, our research is aligned with the current trend of adopting statistical and AI-based methods for attack detection (Hero et al., 2023).

3. **Solution to a Real-world Challenge:** ICPS are the backbone of the fourth industrial revolution. Massive interconnection and digitization of cyber-physical components pose a risk of disrupted and degraded operations due to DDoS attacks (Chapter. 2). The resource limitation of these systems needs tailored security measures to combat DDoS attacks. The security of these resource-constrained ICPS is not just a technical problem but a safeguard for public safety and financial resilience. Addressing the challenge of ensuring the availability of resource-constrained ICPS using light-weight active security measures (Chapters. 4, 8, 10) is a real-world concern that

requires urgent attention (J.-P. A. Yaacoub et al., 2020; Agrawal & Kumar, 2022). Therefore, in this research, we have not only provided the theoretical solutions but have taken one step further by implementing these solutions to demonstrate their practical applicability and effectiveness.

In conclusion, protecting resource-constrained ICPS represents a paradigm shift in cybersecurity. This trend will advance the research capabilities and provide the industries with the solutions to rapidly navigate emerging cyber disruptions.

12.1.3 Threats to Validity

We carefully consider and address the following threats to validity during our research to ensure that the findings are accurate and applied to the broader context.

A threat to ecological validity is a paramount concern to address to ensure the robustness of research findings. It refers whether the findings of the study can be generalised to different situations or setting. The researcher must ensure the study's results can be applied to the real world. In this research, the design and development of LISM as a security solution for resource-constrained ICPS makes our work applicable to the industrial environment. However, as the simulation environment does not replicate the real-world environment completely, the applicability of results on real-world scenarios might be compromised. To mitigate the threat to ecological validity, the proposed solutions were evaluated on different attack scenarios to enhance the diverse simulation scenarios. We selected the publicly available datasets whose network configurations and traffic types mimic the real-world traffic streams. We also did experiments on PLCs and strive to simulate an environment that could closely resemble real-world conditions. In addition, population validity is also a crucial threat to external

validity. It refers to how much study results can be generalised to a larger population. The versatility, scalability, and complexity analysis of our proposed LISM ensure that our solution is a generic solution that can be implemented on various devices in ICPS. For example, the time-complexity of our proposed technique for detecting multi-scale flooding attacks and attack volumes (Chapter. 8) is $O(\alpha log\alpha)$. Our detection technique detects the attack for k samples ($k < n$) in one second, where n is the total number of packets within an attack period t_w. The runtime of computing these samples is α, i.e., $\alpha = t_s \times k$. The complexity of *nlogn* is considered moderate, and scalable more than quadratic or higher order complexity even when the input size grows (D.-Z. Du & Ko, 2011). Our technique's complexity is more efficient and scalable than linear log complexity, resulting in less computational time. ICPS applications process large amounts of traffic in real-time. This complexity ensures efficient processing to meet real-world applications' timing constraints and can easily manage large-size data streams without becoming slow. The time-complexity of our technique is an indication that it can identify the attacks without excessive delays and, thus, is suitable for several real-world applications of ICPS.

Researcher bias may affect the research validity due to the researcher's preferences or expectations. Researcher bias could arise due to the researcher's expertise bias and selection bias. The researcher's expertise bias arises when the researcher interest, knowledge or background influences the selection of the research scope and objectives and research process. While in-depth knowledge of a particular domain is mandatory, a researcher could not be proficient in every aspect of an extensive research area. To minimise the risk of researcher biases, research topics, objectives, scopes and research questions were established and discussed carefully with the consent of subject experts, colleagues, and co-authors to gather feedback, seek advice, and share ideas. The selection bias regarding the number

of relevant studies is also a potential and critical threat to validity. Including a sufficient number of studies is essential to comprehensively understand the research topic and gaps in the current state of knowledge. The scope, novelty of research domains, established research objectives, or clear separation between concepts help to determine the size of the set of primary studies. Thus, the primary studies were chosen carefully to include only relevant works through a comprehensive review of each study's complete text, experimental scenarios, and case studies. In case of uncertainties, consultations were conducted with domain experts. Also, the interaction effect could be an internal threat that introduces complexity and bias and needs careful consideration. The interaction effect could be observed when the effectiveness of a system or component in identifying one type of attack could impact its performance in identifying another type of attack. Incorporating more DDoS detection algorithms in LISM or using the concurrent mode of communication might expose LISM to this threat. In interaction effect, identifying several types of attacks through LISM could introduce ambiguity in analysing the observed effect, whether the observed outcomes are due to one type of attack or the combined impact. This threat could be mitigated in the future if more algorithms will be introduced in LISM.

12.2 Future Directions

12.2.1 Light-weight (D)DoS attacks mitigation strategies

Future development could focus on the designing and developing light-weight DoS/DDoS attack mitigation strategies that align with the inherent characteristics of resource-constrained ICPS. In the literature, various DoS/DDoS attack mitigation mechanisms like rate limiting, packet marking, blocking ports, traceback,

acknowledgement, trust-based methods, ML-based solutions, and statistics-based solutions are available (Alhijawi, Almajali, Elgala, Salameh & Ayyash, 2022; Bashendy, Tantawy & Erradi, 2023; Rios et al., 2022; J. Singh & Behal, 2020; Bakr, El-Aziz & Hefny, 2019; Somani, Gaur, Sanghi, Conti & Buyya, 2017; Mahjabin et al., 2017; Kordestani & Saif, 2021; Imran, Durad, Khan & Derhab, 2019; Ortega-Fernandez & Liberati, 2023). These solutions are common security measures, and their practical implications pose significant limitations when applied to resource-constrained ICPS. These existing mechanisms are complex to configure, could increase energy consumption, introduce performance overheads, and lead to false positives. In general, there is a lack of light-weight solutions which can be employed directly to the resource constrained ICPS to mitigate the consequences of DoS/DDoS attacks. These research gaps indicate a need for an architectural-based solution that can align with the constraints of resource-constrained systems. The architectural solutions involve the mitigation strategies designed and developed for different layers and/or components of ICPS. The performance evaluation of such solutions needs to be investigated.

12.2.2 Optimisation of Reinforcement learning (RL) model for attack detection in resource-constrained ICPS

Another potential direction is RL optimization, which seems to be a promising approach to improving intrusion detection within resource-constrained ICPS. RL can train the models that continuously adapt and evolve to detect unknown attack patterns in the changing network traffic (Ortega-Fernandez & Liberati, 2023; H. Zhang & Yu, 2020; Suwannalai & Polprasert, 2020; Utic & Ramachandran, 2022; Arshad et al., 2022; Bhutto, Vu, Elmroth, Tay & Bhuyan, 2022). They can also adapt their detection mechanisms based on the available resources, so the capacities

of resource-constrained devices are allocated only for optimal attack detection. However, RL is resource-intensive and needs extensive data to learn policies effectively (Arshad et al., 2022; Suwannalai & Polprasert, 2020). Therefore, the direct implementation of RL in the resource-constrained environment is challenging. This crucial issue highlights a need for more efficient online learning strategies to adapt the RL algorithm for timely attack detection in resource-constrained ICPS.

12.2.3 Reverse Engineering: A valuable approach against DDoS attacks

Reverse engineering could be used in detecting and mitigating DDoS attacks. Reverse engineering can help understand the attack vectors, vulnerability identifications, and determination of attack patterns and signatures (Breier, Jap, Hou, Bhasin & Liu, 2021; Fernández-Caramés, Fraga-Lamas, Suárez-Albela & Castedo, 2016; Geng et al., 2023). Resource-constrained devices lack the computational power for complex analysis. They do not have a memory for storing and processing extensive signature databases. Using reverse engineering, a lightweight, signature-free DDoS attack detection and mitigation algorithm can be proposed. These optimised algorithms could help to protect the resource-constrained devices without relying on predefined signatures of attack.

12.2.4 Enhancing resource-constrained ICPS security using Transfer learning

Resource-constrained ICPS are commonly used for real-time operations in the industrial environment. The protection of these systems against DDoS attacks is very crucial. Transfer learning can be applied to enhance the security of resource-constrained ICPS. The pre-trained model used in traditional networks can be used

for attack detection in ICPS using transfer learning (Zhao, Shetty, Pan, Kamhoua & Kwiat, 2019; Catillo, Del Vecchio, Pecchia & Villano, 2022; Shafiq et al., 2022; W. Wang et al., 2021). Many existing pre-trained models have extensive memory and storage requirements. Due to computation overhead, these models cannot be deployed in resource-constrained ICPS. Moreover, fine-tuning a pre-trained model for the resource-constrained environment is challenging and time-consuming. In order to make a real-time detection, there is a need to overcome these challenges by optimising the pre-trained models. The optimised pre-trained models can be utilised in resource-constrained ICPS for attack detection effectively and efficiently.

12.2.5 Resource-efficient network forensic analysis for DDoS attack detection and mitigation

Network forensic analysis can be used to detect and mitigate the attacks on resource-constrained ICPS proactively. Network forensics is a new and evolving branch of digital forensics that deals with the proofs related to network traffic patterns in the form of logs and network flows (Moustafa & Slay, 2018; Fadlil, Riadi & Aji, 2017; Volarević, Tomić & Milohanić, 2022). It can uncover the attack details and its source. It can identify the sudden increase in incoming traffic or a high volume of requests. Mitigation strategies can be employed to minimise the malicious traffic's impact.

12.2.6 A comprehensive integrity and confidentiality attack taxonomy for ICPS

In addition to the availability attacks, we can extend our taxonomy for integrity and confidentiality attacks, like eavesdropping, side-channel attacks, or command injection attacks (Duo, Zhou & Abusorrah, 2022; Kayan et al., 2021). These

attacks are also considered to be significant threats to the resource-constrained ICPS. A comprehensive taxonomy is essential to understand, analyse, and mitigate these security threats on ICPS. This taxonomy can help security professionals and policymakers to understand the nature of such threats in ICPS. It could help in development and implementation of the appropriate security controls to protect these critical systems.

12.2.7 Representation of real-time processing times of DDoS attack detection techniques

The hard real-time processing time could be represented using various methods, including timing diagrams, Gantt charts, flow charts, and schedulability analyses using mathematical models. These methods show the timing constraints and deadlines a system must meet to function correctly. For example, the timing diagrams show various event sequences and associated deadlines. Similarly, Gantt Charts help visualise the task's start, end, slack, and overlapped periods. In future work, we can use any of these methods to robustly represent the processing times taken by our proposed techniques to detect the attacks.

12.2.8 Adversarial threats and their impact on OSELM

Adversarial threats can lead to incorrect classification or predictions by altering training data or extracting useful information about training data with model inversion (Alotaibi & Rassam, 2023; A. Chakraborty, Alam, Dey, Chattopadhyay & Mukhopadhyay, 2021; Ibitoye, Abou-Khamis, Matrawy & Shafiq, 2019; A. Chakraborty, Alam, Dey, Chattopadhyay & Mukhopadhyay, 2018; Ibitoye, Shafiq & Matrawy, 2019). Our proposed optimised OSELM-based slow-rate attack detection framework can be under the threat of such attacks. There is a

need for some light-weight algorithms, like robust optimization algorithms, which can improve the robustness and the model stability. In the future, the robust optimisation algorithm could be used to find out the parameters of the optimised OSELM model, which have minimal sensitivity to adversarial perturbations. The parameters are adjusted to reduce the expected loss within the defined uncertainty sets. These sets specify the variation range the model can handle and can be part of the online learning process. This online learning model parameter adjustment using a robust optimisation algorithm enables the model to adapt continuously against evolving adversarial threats.

12.3 Conclusions

This research was primarily focused on providing active security for the self-protection of resource-constrained ICPS. In this research, we have followed multi-phased design science research framework and establish several research objectives discussed in Section. 1.2.1.

In order to identify the critical challenges and current research directions within ICPS (RO1), a comprehensive SMS was conducted in Chapter 2. The primary studies of SMS employed various tools, techniques, formalism methods, and frameworks to improve the security, performance, and dependability of ICPS. The primary studies were classified based on various application domains, requirements engineering activities, and industrial standards. We contextualise the findings of SMS through the conceptual model. The insights attained from the SMS lead to a notable observation that more attention should be given to dealing with DoS/DDoS cyber attacks that significantly impact the availability of components within ICPS.

The second research objective (RO2) was to perform a comprehensive SLR of

existing DoS/DDoS attack detection techniques in ICPS (Chapter 4). The results of SLR indicate that resource-constrained ICPS need light-weight active security mechanism to detect the DoS/DDoS attacks without burdening the computational resources. Also, analysing the abnormal behaviour of a system in a time domain poses a significant challenge. The time domain analysis do not separate harmonics, unusual spikes, and noise that appear as random signal variations and are identified as attack signals. Considering the identified gaps, a novel light-weight active security approach was proposed in the frequency domain discussed in Chapter. 4 to address research objective 4 (RO4). Our approach detects DoS/DDoS attacks by comparing a frequency signature of the incoming flow of packets with a baseline. We took advantage of a flexible and light-weight time-to-frequency transformation approach. Our prototype implementation and evaluation (RO5) show that the proposed active security solution is light-weight and suitable for attack detection in resource-constrained ICPS platforms with minimal computational cost.

Chapter 6 addresses the third research objective (RO3) by presenting the novel proposed taxonomy for categorising widely recognised (distributed) denial of service attacks, emphasising multi-vector and cross-domain attacks. More than fifty denial-of-service attacks on smart manufacturing systems were categorised as Endpoint and Network (distributed) Denial of Service attacks utilising the taxonomy. The taxonomy criteria helped us identify the multi-vectors utilised by the attackers to launch unavailability, delayed, manipulation, buffer overflow, amplification, jamming or direct flooding attacks. These attacks degrade or disrupt the availability of devices or their services or exhaust their bandwidth.

The fourth research objective (RO4) has been achieved by designing and developing the light-weight technique for detecting multi-scale flooding attacks and attack volumes, presented in Chapter 8. The multi-scale flooding attacks are detected using statistically robust and resource-efficient multi-methods. These

methods identify the attacks by determining the randomness in the incoming traffic's frequency signature and the similarity of the incoming traffic's frequency profile with the baseline. The dissimilarity metric is used to identify the strength of the attack (attack volumes). The experimental results and performance criteria indicate that our technique represents a step towards securing resource-constrained ICPS.

In the final iteration of research objective 4 (RO4), Chapter 10 presents an optimised OSELM-based novel light-weight active security framework for binary and multi-class SRA detection. The memory and space requirements of OSELM model were reduced using optimisation techniques. An introduction of a straightforward training method and optimised OSELM are the foundation of the proposed framework. We have evaluated the effectiveness of our proposed framework on PLC-based ICPS. The reduced attack detection time and CPU and memory overhead improvements indicate that our light-weight framework could provide active security to resource constrained ICPS.

Implementing LISM for offering active security to resource-constrained ICPS with the detection of DDoS attacks improves the discipline and state-of-the-art, benefiting various stakeholders. Attack detection is an initial layer of security; however, the comprehensive cybersecurity strategy also includes developing mechanisms to mitigate the identified attacks. Therefore, to defend resource-constrained ICPS, a holistic approach is required to combine detection and mitigation strategies. The attack mitigation strategy that aligns with the inherent characteristics of resource-constrained ICPS could be considered a future development. Also, cybersecurity objectives include confidentiality and integrity in addition to the availability. The proposed light-weight solutions can be extended to incorporate attacks that impact the confidentiality and integrity of resource-constrained ICPS.

References

Aamir, M. & Zaidi, S. M. A. (2021). Clustering based semi-supervised machine learning for ddos attack classification. *Journal of King Saud University-Computer and Information Sciences*, *33*(4), 436–446.

Aceto, L., Ingólfsdóttir, A., Larsen, K. G. & Srba, J. (2007). *Reactive systems: modelling, specification and verification* (1st ed.). Cambridge University Press.

Adepu, S., Kang, E., Jackson, D. & Mathur, A. (2016, 05). Model-based security analysis of a water treatment system. In *2nd international workshop on software engineering for smart cyber-physical systems (sescps).* Austin, Texas. doi: https://doi.org/10.1145/2897035.2897041

Adepu, S. & Mathur, A. (2016a, 01). Introducing cyber security at the design stage of public infrastructures: a procedure and case study. In *Complex systems design & management asia* (Vol. 426, pp. 75–94). Cham: Springer. doi: https://doi.org/10.1007/978-3-319-29643-2_6

Adepu, S. & Mathur, A. (2016b, 01). An investigation into the response of a water treatment system to cyber attacks. In *Ieee 17th international symposium on high assurance systems engineering (hase).* Orlando, United States. doi: https://doi.org/10.1109/HASE.2016.14

Adepu, S. & Mathur, A. (2016c, 05). Using process invariants to detect cyber attacks on a water treatment system. In *Ifip international conference on information security and privacy protection* (pp. 91–104). Gent, Belgium. doi: https://doi.org/10.1007/978-3-319-33630-5_7

Agrawal, N. & Kumar, R. (2022). Security perspective analysis of industrial cyber physical systems (I-CPS): A decade-wide survey. *ISA transactions*, *130*, 10–24.

Ahmad, E., Dong, Y., Larson, B., Lü, J., Tang, T. & Zhan, N. (2015). Behavior modeling and verification of movement authority scenario of Chinese train control system using AADL. *Science China Information Sciences*, *58*(11), 1–20.

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J. & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.

Ahmadi, A., Sodhro, A. H., Cherifi, C., Cheutet, V. & Ouzrout, Y. (2018).

Evolution of 3C cyber-physical systems architecture for Industry 4.0. In *International workshop on service orientation in holonic and multi-agent manufacturing* (pp. 448–459).

Ahmed, R. & Robinson, S. (2007). Simulation in business and industry: how simulation context can affect simulation practice? In *Proceedings of the 2007 spring simulation multiconference-volume 3* (pp. 152–159). Virginia, USA.

Akella, R. & McMillin, B. M. (2009). Model-checking BNDC properties in cyber-physical systems. In *2009 33rd annual ieee international computer software and applications conference* (Vol. 1, pp. 660–663). Seattle, Washington, USA.

Akkaya, I., Derler, P., Emoto, S. & Lee, E. A. (2016). Systems engineering for industrial cyber–physical systems using aspects. *Proceedings of the IEEE*, *104*(5), 997–1012.

Alexander, G., Oleg, K. & Yaroslav, S. (2022). *DDoS attacks in Q4 2021.* `https://securelist.com/ddos-attacks-in-q4-2021/105784/`. ([Online; accessed 15-February-2022])

Al-Haija, Q. A., Altamimi, S. & AlWadi, M. (2024). Analysis of extreme learning machines (elms) for intelligent intrusion detection systems: A survey. *Expert Systems with Applications*, 124317.

Alhijawi, B., Almajali, S., Elgala, H., Salameh, H. B. & Ayyash, M. (2022). A survey on dos/ddos mitigation techniques in sdns: Classification, comparison, solutions, testing tools and datasets. *Computers and Electrical Engineering*, *99*, 107706.

Ali, B. H., Sulaiman, N., Al-Haddad, S. A. R., Atan, R., Hassan, S. L. M. & Alghrairi, M. (2021). Identification of distributed denial of services anomalies by using combination of entropy and sequential probabilities ratio test methods. *Sensors*, *21*(19). doi: 10.3390/s21196453

Alotaibi, A. & Rassam, M. A. (2023). Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. *Future Internet*, *15*(2), 62.

Althobaiti, M. M., Kumar, K. P. M., Gupta, D., Kumar, S. & Mansour, R. F. (2021). An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems. *Measurement*, *186*, 110145.

Amidan, B. G., Ferryman, T. A. & Cooley, S. K. (2005). Data outlier detection using the chebyshev theorem. In *2005 ieee aerospace conference* (pp. 3814–3819).

Anderson, C. (2015, may). Docker [software engineering]. *IEEE Software*, *32*(03), 102-c3. doi: 10.1109/MS.2015.62

Ansari, M. F., Dash, B., Sharma, P. & Yathiraju, N. (2022). The impact and limitations of artificial intelligence in cybersecurity: A literature review. *International Journal of Advanced Research in Computer and Communication Engineering*.

Arshad, K., Ali, R. F., Muneer, A., Aziz, I. A., Naseer, S., Khan, N. S. &

Taib, S. M. (2022). Deep reinforcement learning for anomaly detection: A systematic review. *IEEE Access*.

Asad, M., Asim, M., Javed, T., Beg, M. O., Mujtaba, H. & Abbas, S. (2020). Deepdetect: detection of distributed denial of service attacks using deep learning. *The Computer Journal*, *63*(7), 983–994.

Ashok, A., Govindarasu, M. & Wang, J. (2017). Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid. *Proceedings of the IEEE*, *105*(7), 1389–1407.

Askarpour, M., Ghezzi, C., Mandrioli, D., Rossi, M. & Tsigkanos, C. (2019). Formal methods in designing critical cyber-physical systems. In *From software engineering to formal methods and tools, and back* (Vol. 11865, pp. 110–130). Porto, Portugal: Springer. doi: https://doi.org/10.1007/978-3-030-30985-5_8

BA, K. & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Tech. Rep.). The Pennsylvania State University: Keele University and Durham University Joint Report.

Bae, K., Krisiloff, J., Meseguer, J. & Ölveczky, P. (2015, 06). Designing and verifying distributed cyber-physical systems using multirate pals: an airplane turning control system case study. *Science of Computer Programming*, *103*, 13–50. doi: https://doi.org/10.1016/j.scico.2014.09.011

Bakr, A., El-Aziz, A. & Hefny, H. A. (2019). A survey on mitigation techniques against ddos attacks on cloud computing architecture. *International Journal of Advanced Science and Technology*, *28*(12), 187–200.

Balasubramaniyan, S., Srinivasan, S., Buonopane, F., Subathra, B., Vain, J. & Ramaswamy, S. (2016). Design and verification of cyber-physical systems using truetime, evolutionary optimization and uppaal. *Microprocessors and Microsystems*, *42*, 37–48.

Bartocci, E., Manjunath, N., Mariani, L., Mateis, C., Ničković, D. & Pastore, F. (2020). CPSDebug: a tool for explanation of failures in cyber-physical systems. In *Proceedings of the 29th acm sigsoft international symposium on software testing and analysis* (p. 569-–572). New York, NY, USA. doi: https://doi.org/10.1145/3395363.3404369

Bashendy, M., Tantawy, A. & Erradi, A. (2023). Intrusion response systems for cyber-physical systems: A comprehensive survey. *Computers & Security*, *124*, 102984.

Behal, S., Kumar, K. & Sachdeva, M. (2018). D-face: An anomaly based distributed approach for early detection of ddos attacks and flash events. *Journal of Network and Computer Applications*, *111*, 49–63.

Bernardi, S., Gentile, U., Marrone, S., Merseguer, J. & Nardone, R. (2020). Security modelling and formal verification of survivability properties: application to cyber–physical systems. *Journal of Systems and Software*, 110–746. (accessed on 22 October 2020)

Bhardwaj, A., Subrahmanyam, G., Avasthi, V., Sastry, H. & Goundar, S. (2016). DDoS Attacks, New DDoS Taxonomy and Mitigation Solutions – A Survey.

In *Ieee international conf. on signal processing, communication, power and embedded system (scopes)* (pp. 793–798).

Bhatia, S., Behal, S., Ahmed, I., Somani, G. & Poovendran, R. (2018). Distributed denial of service attacks and defense mechanisms: Current landscape and future directions. In *Versatile cybersecurity* (pp. 55–97). Cham: Springer International Publishing.

Bhutto, A. B., Vu, X. S., Elmroth, E., Tay, W. P. & Bhuyan, M. (2022). Reinforced transformer learning for vsi-ddos detection in edge clouds. *IEEE Access*, *10*, 94677–94690.

Biron, Z. A., Dey, S. & Pisu, P. (2018). Real-time detection and estimation of denial of service attack in connected vehicle systems. *IEEE Transactions on Intelligent Transportation Systems*, *19*(12), 3893–3902.

Bourque, P. & Fairley, R. E. (2014). *Guide to the software engineering body of knowledge (swebok (r)): version 3.0*. IEEE Computer Society Press.

Bouskela, D. & Jardin, A. (2018). Etl: a new temporal language for the verification of cyber-physical systems. In *Annual ieee international systems conference (syscon)* (pp. 1–8). Vancouver, BC, Canada. doi: https://doi.org/10.1109/SYSCON.2018.8369502

Bouskela, D., Nguyen, T. & Jardin, A. (2017). Toward a rigorous approach for verifying cyber-physical systems against requirements. *Canadian Journal of Electrical and Computer Engineering*, *40*(2), 66–73.

Bray, T. (2017, December). *The JavaScript Object Notation (JSON) data interchange format* (No. 8259). RFC 8259. RFC Editor. Retrieved from `https://rfc-editor.org/rfc/rfc8259.txt` doi: https://doi.org/10.17487/RFC8259

Breier, J., Jap, D., Hou, X., Bhasin, S. & Liu, Y. (2021). Sniff: reverse engineering of neural networks with fault attacks. *IEEE Transactions on Reliability*, *71*(4), 1527–1539.

Bu, L., Wang, Q., Chen, X., Wang, L., Zhang, T., Zhao, J. & Li, X. (2011). Toward online hybrid systems model checking of cyber-physical systems' time-bounded short-run behavior. *ACM SIGBED Review*, *8*(2), 7–10.

Cao, L., Jiang, X., Zhao, Y., Wang, S., You, D. & Xu, X. (2020). A survey of network attacks on cyber-physical systems. *IEEE Access*, *8*, 1-8.

Carvalho, L. K., Wu, Y.-C., Kwong, R. & Lafortune, S. (2018). Detection and mitigation of classes of attacks in supervisory control systems. *Automatica*, *97*, 121–133.

Catillo, M., Del Vecchio, A., Pecchia, A. & Villano, U. (2022). Transferability of machine learning models learned from public intrusion detection datasets: the cicids2017 case study. *Software Quality Journal*, *30*(4), 955–981.

Cengic, G. & Akesson, K. (2010). On formal analysis of IEC 61499 applications, part b: Execution semantics. *IEEE Transactions on Industrial Informatics*, *6*(2), 145-154. doi: https://doi.org/10.1109/TII.2010.2040393

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, *1*(2), 1.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. & Mukhopadhyay, D. (2018). Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. & Mukhopadhyay, D. (2021). A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, *6*(1), 25–45.

Chakraborty, C., Nagarajan, S. M., Devarajan, G. G., Ramana, T. V. & Mohanty, R. (2023). Intelligent ai-based healthcare cyber security system using multi-source transfer learning method.
doi: 10.1145/3597210

Chebudie, A. B., Minerva, R. & Rotondi, D. (2015). *Towards a definition of the internet of things (iot).* IEEE.

Chelladhurai, J., Chelliah, P. R. & Kumar, S. A. (2016). Securing docker containers from denial of service (dos) attacks. In *2016 ieee international conference on services computing (scc)* (p. 856-859). doi: 10.1109/SCC.2016.123

Chen, B., Pattanaik, N., Goulart, A., Butler-purry, K. L. & Kundur, D. (2015). Implementing attacks for modbus/tcp protocol in a real-time cyber physical system test bed. In *Ieee int'l workshop technical committee on communications quality and reliability (cqr)* (p. 1-6). doi: 10.1109/CQR.2015.7129084

Chen, M., Chen, J., Wei, X. & Chen, B. (2021). Is low-rate distributed denial of service a great threat to the internet? *IET Information Security*, *15*(5), 351–363.

Chen, Q., Abdelwahed, S. & Erradi, A. (2013). A model-based approach to self-protection in computing system. In *Proceedings of the 2013 acm cloud and autonomic computing conference* (pp. 1–10).

Chen, Y., Dai, W., Zhang, Z., Pang, C. & Vyatkin, V. (2018). A case study on knowledge driven code generation for software-defined industrial cyber-physical systems. In *Iecon 2018-44th annual conference of the ieee industrial electronics society* (pp. 4687–4692). Washington, DC, USA.

Chen, Y. & Hwang, K. (2007). Spectral analysis of tcp flows for defense against reduction-of-quality attacks. In *2007 ieee international conference on communications* (p. 1203-1210). doi: 10.1109/ICC.2007.204

Chowdhury, A. & A Raut, S. (2019). Benefits, challenges, and opportunities in adoption of industrial iot. *International Journal of Computational Intelligence & IoT*, *2*(4).

Clarke, E. M. & Zuliani, P. (2011). Statistical model checking for cyber-physical systems. In *International symposium on automated technology for verification and analysis* (pp. 1–12). Taipei, Taiwan.

Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J. & Talcott, C. (2007). *All about maude - a high-performance logical framework: how to specify, program and verify systems in rewriting logic.* Berlin, Heidelberg: Springer-Verlag.

Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., ... Cohen, I. (2009). Pearson correlation coefficient. *Noise reduction in speech processing*,

1–4.

Colombo, A. W., Bangemann, T., Karnouskos, S., Delsing, J., Stluka, P., Harrison, R., ... et al (2014). Industrial cloud-based cyber-physical systems. *The IMC-AESOP Approach*, *22*, 4–5. doi: https://doi.org/10.1007/978-3-319-05624-1

Colombo, A. W., Karnouskos, S., Kaynak, O., Shi, Y. & Yin, S. (2017). Industrial cyberphysical systems: a backbone of the fourth industrial revolution. *IEEE Industrial Electronics Magazine*, *11*(1), 6–16. doi: https://doi.org/10.1109/MIE.2017.2648857

Cover Thomas, M. & Thomas Joy, A. (1991). Elements of information theory. *New York: Wiley*, *3*, 37–38.

Dadkhah, S., Mahdikhani, H., Danso, P. K., Zohourian, A., Truong, K. A. & Ghorbani, A. A. (2022). Towards the development of a realistic multidimensional iot profiling dataset. In *2022 19th annual international conference on privacy, security and trust (pst)* (p. 1-11). doi: 10.1109/PST55820.2022.9851966

Dang, T., Mady, A. E.-D., Boubekeur, M., Kumar, R. & Moulin, M. (2016). Validation of industrial cyber-physical systems: an application to hvac systems. In *International conference on complex systems design & management* (pp. 57–69). Paris, France.

David, J. & Thomas, C. (2015). Ddos attack detection using fast entropy approach on flow-based network traffic. *Procedia Computer Science*, *50*, 30–36.

David, J. & Thomas, C. (2019). Efficient ddos flood attack detection using dynamic thresholding on flow-based network traffic. *Computers & Security*, *82*, 284–295.

David, J. & Thomas, C. (2020). Detection of distributed denial of service attacks based on information theoretic approach in time series models. *Journal of Information Security and Applications*, *55*, 102621.

Davis, J. A., Clark, M., Cofer, D., Fifarek, A., Hinchman, J., Hoffman, J., ... Wagner, L. (2013). Study on the barriers to the industrial adoption of formal methods. In *International workshop on formal methods for industrial critical systems* (pp. 63–77). Madrid, Spain.

Denno, P. O. & Blackburn, M. (2014). Virtual design and verification of cyber physical systems: industrial process plant design. In *Conference on systems engineering research (cser 2014)*. CA, USA.

Derigent, W., Cardin, O. & Trentesaux, D. (2020). Industry 4.0: contributions of holonic manufacturing control architectures and future challenges. *Journal of Intelligent Manufacturing*, 1–22. doi: https://doi.org/10.1007/s10845-020-01532-x

Diaba, S. Y., Shafie-khah, M. & Elmusrati, M. (2022). On the performance metrics for cyber-physical attack detection in smart grid. *Soft Computing*, *26*(23), 13109–13118.

Ding, D., Han, Q.-L., Xiang, Y., Ge, X. & Zhang, X.-M. (2018, January). A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomput.*, *275*(C), 1674–1683. Retrieved from `https://doi.org/10`

.1016/j.neucom.2017.10.009 doi: 10.1016/j.neucom.2017.10.009

Drozdov, D., Patil, S., Dubinin, V. & Vyatkin, V. (2019). Towards formal ASM semantics of timed control systems for industrial CPS. In *24th ieee international conference on emerging technologies and factory automation (etfa)* (pp. 1682–1685). Zaragoza, Spain.

Drozdov, D., Patil, S. & Vyatkin, V. (2017). Formal modelling of distributed automation cps with cp-agnostic software. *Service Orientation in Holonic and Multi-Agent Manufacturing*, 35.

Du, D., Huang, P., Jiang, K. & Mallet, F. (2018). pCSSL: a stochastic extension to MARTE/CCSL for modeling uncertainty in cyber physical systems. *Science of Computer Programming*, *166*, 71–88.

Du, D.-Z. & Ko, K.-I. (2011). *Theory of computational complexity* (Vol. 58). John Wiley & Sons.

Duo, W., Zhou, M. & Abusorrah, A. (2022). A survey of cyber attacks on cyber physical systems: Recent advances and challenges. *IEEE/CAA Journal of Automatica Sinica*, *9*(5), 784–800.

Dyba, T., Dingsoyr, T. & Hanssen, G. K. (2007). Applying systematic reviews to diverse study types: an experience report. In *First international symposium on empirical software engineering and measurement (esem 2007)* (pp. 225–234). NW Washington, DC,United States.

Eduard, K. (n.d.). *Library Flaw Could Crash HART-Based ICS Field Devices.* `https://www.securityweek.com/library-flaw-could-crash-hart-based-ics-field-devices`. ([Online; accessed 23-February-2022])

ElDahshan, K. A., AlHabshy, A. A. & Hameed, B. I. (2022). Meta-heuristic optimization algorithm-based hierarchical intrusion detection system. *Computers*, *11*(12), 170.

Elgendi, I., Hossain, M. F., Jamalipour, A. & Munasinghe, K. S. (2019). Protecting cyber physical systems using a learned mape-k model. *IEEE Access*, *7*, 90954–90963.

Elleithy, K. M., Blagovic, D., Cheng, W. K. & Sideleau, P. (2005). Denial of service attack techniques: analysis, implementation and comparison. *Journal of Systemics, Cybernetics, and Informatics*, *3*(1), 66-71.

Elmqvist, H., Boudaud, F., Broenink, J., Brück, D., Ernst, T., Fritzson, P., ... Mattsson, S. (1999). ModelicaTM-a unified object-oriented language for physical systems modeling. *Tutorial and Rationale, versión*, *1*.

Emad, F. (2023). *How DDoS attacks impact business continuity.* `https://www.edgemiddleeast.com/security/how-ddos-attacks-impact-business-continuity`. ([Online; accessed 10-Novemeber-2023])

Erhan, D. & Anarim, E. (2020, 08). Boğaziçi university distributed denial of service dataset. *Data in Brief*, *32*, 106187. doi: 10.1016/j.dib.2020.106187

Ezio, B., N, M., L, M., Cristinel, M. & D, N. (2019). Automatic failure explanation in CPS models. In *17th international conference on software engineering and formal methods* (Vol. 11724, pp. 69–86). Oslo, Norway. doi: https://doi.org/10.1007/978-3-030-30446-1_4

Fadlil, A., Riadi, I. & Aji, S. (2017). Review of detection ddos attack detection using naive bayes classifier for network forensics. *Bulletin of Electrical Engineering and Informatics*, *6*(2), 140–148.

Faisal, M. S. B., Habib, A., Hossain, M. A., Rashid, C. S. & Nandi, D. (2021). Investigation of security challenges from the perspective of stakeholders in iot. *AIUB Journal of Science and Engineering (AJSE)*, *20*(2), 8–19.

Famili, A., Shen, W.-M., Weber, R. & Simoudis, E. (1997). Data preprocessing and intelligent data analysis. *Intelligent data analysis*, *1*(1), 3–23.

Farhan, B. I. & Jasim, A. D. (2022). Performance analysis of intrusion detection for deep learning model based on cse-cic-ids2018 dataset. *Indonesian Journal of Electrical Engineering and Computer Science*, *26*(2), 1165–1172.

Fawzi, H., Tabuada, P. & Diggavi, S. (2014). Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic control*, *59*(6), 1454–1467.

Feiler, P. H. & Gluch, D. P. (2012). *Model-based engineering with aadl: an introduction to the sae architecture analysis & design language*. Addison-Wesley.

Felizardo, K. R., Nakagawa, E. Y., Feitosa, D., Minghim, R. & Maldonado, J. C. (2010). An approach based on visual text mining to support categorization and classification in the systematic mapping. In *14th international conference on evaluation and assessment in software engineering (ease)* (pp. 1–10). Swindon,United Kingdom.

Fernández-Caramés, T. M., Fraga-Lamas, P., Suárez-Albela, M. & Castedo, L. (2016). Reverse engineering and security evaluation of commercial tags for rfid-based iot applications. *Sensors*, *17*(1), 28.

Ferrante, O., Di Guglielmo, L., Senni, V. & Ferrari, A. (2017). Application of model-based safety assessment to the validation of avionic electrical power systems. In *International symposium on model-based safety and assessment* (pp. 243–254). Trento, Italy.

Fink, G. A., Edgar, T. W., Rice, T. R., MacDonald, D. G. & Crawford, C. E. (2017). Security and privacy in cyber-physical systems. In *Cyber-physical systems* (pp. 1–23). Boston: Academic Press. doi: https://doi.org/10.1016/B978-0-12-803801-7.00009-2

Fisher, A., Jacobson, C. A., Lee, E. A., Murray, R. M., Sangiovanni-Vincentelli, A. & Scholte, E. (2014). Industrial cyber-physical systems – iCyPhy. In *Proceedings of the fourth international conference on complex systems design & management* (pp. 21–37). France, Paris. doi: https://doi.org/10.1007/978-3-319-02812-5_2

Fouladi, R., Ermis, O. & Anarim, E. (2019). Anomaly-based ddos attack detection by using sparse coding and frequency domain. *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 1-6.

France, R., Evans, A., Lano, K. & Rumpe, B. (1998). The UML as a formal modeling notation. *Computer Standards & Interfaces*, *19*(7), 325–334.

Franceschini, F., Maisano, D. & Mastrogiacomo, L. (2016). Empirical analysis and classification of database errors in Scopus and Web of Science. *Journal of Informetrics*, *10*(4), 933–953. doi: https://doi.org/10.1016/j.joi.2016.07.003

Fu, Y., Duan, X., Wang, K. & Li, B. (2022). Low-rate denial of service attack detection method based on time-frequency characteristics. *Journal of Cloud Computing*, *11*(1), 31.

Fuchs, A., Gürgens, S., Weber, D., Bodenstedt, C. & Ruland, C. (2010). Formalization of smart metering requirements. In *Proceedings of the international workshop on security and dependability for resource constrained embedded systems* (pp. 1–6). Vienna, Austria.

Funchal, G., Zahid, F., Melo, V., Kuo, M. M. Y., Pedrosa, T., Sinha, R., ... Leitão, P. (2023). *Replication Data for: An Intrusion Detection System Dataset for a Multi-Agent Cyber-Physical Conveyor System.* Instituto Politecnico de Bragança. Retrieved from `https://doi.org/10.34620/dadosipb/JLONBO` doi: 10.34620/dadosipb/JLONBO

Function blocks (European Standard). (2013, February). Avenue Marnix, Brussels: British-Standard-Institution.

Gabmeyer, S., Kaufmann, P., Seidl, M., Gogolla, M. & Kappel, G. (2019). A feature-based classification of formal verification techniques for software models. *Software & Systems Modeling*, *18*(1), 473–498.

Gao, J., Chai, S., Zhang, B. & Xia, Y. (2019). Research about dos attack against icps. *Sensors*, *19*(7), 1542.

Garcia, L., Mitsch, S. & Platzer, A. (2019). HyPLC: Hybrid Programmable Logic Controller Program Translation for Verification. In *Proceedings of the 10th acm/ieee international conference on cyber-physical systems* (pp. 47–56). Montreal Quebec, Canada. doi: https://doi.org/10.1145/3302509.3311036

Gavric, Z. & Simic, D. (2018). Overview of DoS attacks on wireless sensor networks and experimental results for simulation of interference attacks. *Ingeniería e Investigación*, *38*(1), 130–138.

Gawanmeh, A., Alwadi, A. & Parvin, S. (2017). Formal verification of control strategies for a cyber physical system. In *Ieee 37th international conference on distributed computing systems workshops (icdcsw)* (pp. 91–96). Atlanta, GA, USA. doi: https://doi.org/10.1109/ICDCSW.2017.59

Geng, Y., Che, X., Ma, R., Wei, Q., Wang, M. & Chen, Y. (2023). Control logic attack detection and forensics through reverse-engineering and verifying plc control applications. *IEEE Internet of Things Journal*.

Geraldes, A., Geretti, L., Bresolin, D., Muradore, R., Fiorini, P., Mattos, L. & Villa, T. (2018, 09). Formal verification of medical CPS: a laser incision case study. *ACM Transactions on Cyber-Physical Systems*, *2*(4), 1–29. doi: https://doi.org/10.1145/3140237

Gogoi, B. & Ahmed, T. (2022). Http low and slow dos attack detection using lstm based deep learning. In *2022 ieee 19th india council international conference (indicon)* (pp. 1–6).

Gomez, F., Aguilera, M., Olsen, S. & Vanfretti, L. (2020, 04). Software

requirements for interoperable and standard-based power system modeling tools. *Simulation Modelling Practice and Theory*, *103*, 102095. doi: https://doi.org/10.1016/j.simpat.2020.102095

Goorden, M., van de Mortel-Fronczak, J., Reniers, M., Fokkink, W. & Rooda, J. (2019). The impact of requirement splitting on the efficiency of supervisory control synthesis. In *International workshop on formal methods for industrial critical systems* (pp. 76–92). Amsterdam, The Netherlands.

Gowripeddi, V. V., Sasirekha, G., Bapat, J. & Das, D. (2023). Digital twin and ontology based ddos attack detection in a smart-factory 4.0. In *2023 international conference on artificial intelligence in information and communication (icaiic)* (pp. 286–291).

Gracia, T. J. H. & García, A. C. (2018). Sustainable smart cities. creating spaces for technological, social and business development. *Boletín Científico de las Ciencias Económico Administrativas del ICEA*, *6*(12). doi: https://doi.org/10.1007/978-3-319-40895-8

Granat, A., Höfken, H. & Schuba, M. (2017). Intrusion detection of the ics protocol ethercat. In *2nd international conf. on computer, network security and communication engineering* (pp. 113–117).

Grobelna, I. (2020). Formal verification of control modules in cyber-physical systems. *Sensors*, *20*(18), 51–54. doi: https://doi.org/10.3390/s20185154

Gräßler, I., Bodden, E., Pottebaum, J., Geismann, J. & Roesmann, D. (2020, 01). Security-oriented fault-tolerance in systems engineering: a conceptual threat modelling approach for cyber-physical production systems. In *Advanced, contemporary control* (Vol. 1196, pp. 1458–1469). Cham: Springer. doi: https://doi.org/10.1007/978-3-030-50936-1_121

Gu, Q., Li, Z. & Han, J. (2012). Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*.

Gunduz, M. Z. & Das, R. (2020). Cyber-security on smart grid: Threats and potential solutions. *Computer networks*, *169*, 107094.

Gunjal, H., Patel, P., Alzhouri, D. F. & Ebrahimi, D. D. (2023). A smart network intrusion detection system for cyber security of industrial iot.

Gunjal, H., Patel, P. & Ebrahimi, D. D. (2022). Smart network intrusion detection system for cyber security of industrial iot.

Guo, H., Sitton, G. & Burrus, C. (1994). The quick discrete fourier transform. In *Proceedings of icassp '94. ieee international conference on acoustics, speech and signal processing* (Vol. iii, p. III/445-III/448 vol.3). doi: 10.1109/ICASSP.1994.389994

Guo, W. (2019). Robust adaptive online sequential extreme learning machine for predicting nonstationary data streams with outliers. *Journal of Algorithms & Computational Technology*, *13*, 1748302619895421.

Gupta, A. & Gupta, N. (2022). *Research methodology*. SBPD Publications.

Gupta, B. B., Chui, K. T., Arya, V. & Gaurav, A. (2022). A novel approach of securing medical cyber physical systems (mcps) from ddos attacks. In C.-H. Hsu, M. Xu, H. Cao, H. Baghban & A. B. M. Shawkat Ali (Eds.), *Big*

data intelligence and computing (pp. 155–165). Singapore: Springer Nature Singapore.

Guttag, J. V., Horning, J. J., Garland, S., Jones, K., Modet, A. & Wing, J. (1993). *Larch: languages and tools for formal specification*. Springer.

Gyamfi, E. & Jurcut, A. (2022). M-tads: A multi-trust dos attack detection system for mec-enabled industrial lot. In *2022 ieee 27th international workshop on computer aided modeling and design of communication links and networks (camad)* (pp. 166–172).

Hachicha, M., Halima, R. B. & Kacem, A. H. (2019). Formal verification approaches of self-adaptive systems: a survey. *Procedia Computer Science*, *159*, 1853–1862. doi: https://doi.org/10.1016/j.procs.2019.09.357

Haghighi, M. S., Farivar, F., Jolfaei, A. & Tadayon, M. H. (2020). Intelligent robust control for cyber-physical systems of rotary gantry type under denial of service attack. *The Journal of Supercomputing*, *76*(4), 3063–3085.

Hall, A. (2005). Realising the benefits of formal methods. In *7th international conference on formal engineering methods, icfem 2005.* Berlin, Heidelberg. doi: https://doi.org/10.1007/11576280_1

Hand, R., Ton, M. & Keller, E. (2013). Active security. In *Proceedings of the twelfth acm workshop on hot topics in networks.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2535771.2535794

Hao, W., Yang, T. & Yang, Q. (2021). Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber-physical systems. *IEEE Transactions on Automation Science and Engineering*.

Hasan, M., Islam, M. M., Zarif, M. I. I. & Hashem, M. (2019). Attack and anomaly detection in iot sensors in iot sites using machine learning approaches. *Internet of Things*, *7*, 100059.

He, X., Papadopoulos, C., Heidemann, J., Mitra, U., Riaz, U. & Hussain, A. (2005). Spectral analysis of bottleneck traffic. *University of Southern California, Department of Computer Science, Tech. Rep. USC-CSDTR-05-854*.

Hellinger, A., Translation, H. S., Macfarlane, J., Services, B. & Galloway, H. (2011). Cyber-physical Systems. Driving Force for Innovation in Mobility, Health, Energy and Production Acatech (ed.)..

Hero, A., Kar, S., Moura, J., Neil, J., Poor, H. V., Turcotte, M. & Xi, B. (2023, jan 26). Statistics and Data Science for Cybersecurity. *Harvard Data Science Review*, *5*(1). (https://hdsr.mitpress.mit.edu/pub/koyzu1te)

Hissam, S. A., Chaki, S. & Moreno, G. A. (2015). High assurance for distributed cyber physical systems. In *Proceedings of the 2015 european conference on software architecture workshops* (pp. 1–4). Dubrovnik Cavtat, Croatia.

Hnamte, V. & Hussain, J. (2023). Dcnnbilstm: An efficient hybrid deep learning-based intrusion detection system. *Telematics and Informatics Reports*, *10*, 100053. Retrieved from `https://www.sciencedirect.com/science/article/pii/S2772503023000130` doi: https://doi.org/10.1016/j.teler.2023.100053

Hofmann, M. & Klinkenberg, R. (2013). *Rapidminer: data mining use cases*

and business analytics applications. Chapman and Hall/CRC. doi: https://doi.org/10.1201/b16023

Hopcroft, J. E., Motwani, R. & Ullman, J. D. (2000). Introduction to automata theory, languages, and computation, 2nd edition. *SIGACT News*, *32*, 60–65.

Huang, G.-B., Liang, N.-Y., Rong, H.-J., Saratchandran, P. & Sundararajan, N. (2005). On-line sequential extreme learning machine. *Computational Intelligence*, *2005*, 232–237.

Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*(1-3), 489–501.

Huang, J., Zhu, Y., Cheng, B., Lin, C. & Chen, J. (2016). A petrinet-based approach for supporting traceability in cyber-physical manufacturing systems. *Sensors*, *16*(3), 382.

Huang, L., Liang, T. & Kang, E.-Y. (2019). Tool-supported analysis of dynamic and stochastic behaviors in cyber-physical systems. In *Ieee 19th international conference on software quality, reliability and security (qrs)* (pp. 228–239). Sofia, Bulgaria.

Huang, S., Zhou, C.-J., Yang, S.-H. & Qin, Y.-Q. (2015). Cyber-physical system security for networked industrial processes. *International Journal of Automation and Computing*, *12*(6), 567–578.

Hussain, T., Saeed, M. I., Khan, I. U., Aslam, N. & Aljameel, S. S. (2022). Implementation of a clustering-based lddos detection method. *Electronics*, *11*(18), 2804.

Huth, M. & Ryan, M. (2004). *Logic in computer science: modelling and reasoning about systems*. USA: Cambridge University Press.

Ibitoye, O., Abou-Khamis, R., Matrawy, A. & Shafiq, M. O. (2019). The threat of adversarial attacks on machine learning in network security–a survey. *arXiv preprint arXiv:1911.02621*.

Ibitoye, O., Shafiq, O. & Matrawy, A. (2019). Analyzing adversarial attacks against deep learning for intrusion detection in iot networks. In *2019 ieee global communications conference (globecom)* (pp. 1–6).

Iglesias, A., Lu, H., Arellano, C., Yue, T., Ali, S. & Sagardui, G. (2017, 09). Product line engineering of monitoring functionality in industrial cyber-physical systems: a domain analysis. In *Proceedings of the 21st international systems and software product line conference* (pp. 195–204). Sevilla, Spain. doi: https://doi.org/10.1145/3106195.3106223

Imran, M., Durad, M. H., Khan, F. A. & Derhab, A. (2019). Toward an optimal solution against denial of service attacks in software defined networks. *Future Generation Computer Systems*, *92*, 444–453.

Jain, M., Kaur, G. & Saxena, V. (2022). A k-means clustering and svm based hybrid concept drift detection technique for network anomaly detection. *Expert Systems with Applications*, *193*, 116510.

Jalali, S. & Wohlin, C. (2012). Systematic literature studies: database searches vs. backward snowballing. In *Proceedings of the 2012 acm-ieee international symposium on empirical software engineering and measurement* (pp. 29–38).

Lund, Sweden.

JAVA Agent DEvelopment Framework. (n.d.). https://jade.tilab.com/. (Accessed: 2020-02-10)

Jeon, B., Yoon, J.-S., Um, J. & Suh, S.-H. (2020). The architecture development of industry 4.0 compliant smart machine tool system (smts). *Journal of Intelligent Manufacturing*, *31*(8), 1837–1859. doi: https://doi.org/10.1007/s10845-020-01539-4

Jiang, J.-R. (2018). An improved cyber-physical systems architecture for Industry 4.0 smart factories. *Advances in Mechanical Engineering*, *10*(6), 1-15.

John, v., Jan, P. & Richard, B. (2017). *Choosing a design science research methodology* (Vol. 112). ACIS 2017 Proceedings. Retrieved from https://aisel.aisnet.org/acis2017/112 ([Online; accessed 19-September-2023])

Jue, W., Yineng, S., Wu, X. & Dai, W. (2019, 10). A semi-formal requirement modeling pattern for designing industrial cyber-physical systems. In *45th annual conference of the ieee industrial electronics society* (pp. 2883–2888). Lisbon, Portugal. doi: https://doi.org/10.1109/IECON.2019.8926665

Kallel, S. (2011). *Specifying and monitoring non-functional properties* (Unpublished doctoral dissertation). Technische Universität.

Kang, E.-Y., Huang, L. & Mu, D. (2018). Formal verification of energy and timed requirements for a cooperative automotive system. In *Proceedings of the 33rd annual acm symposium on applied computing* (p. 1492–1499). New York, NY, USA. doi: https://doi.org/10.1145/3167132.3167291

Kang, E.-Y., Mu, D., Huang, L. & Lan, Q. (2018). *Model-based verification and validation of an autonomous vehicle system* (Vol. abs/1803.06103). (arXiv preprint, arXiv:1803.06103)

Karter, R. & Hom, N. (2017). *Cybersecurity stakeholders.* https://www.sfmagazine.com/articles/2017/november/cybersecurity-stakeholders/?psso=true. ([Online; accessed 4-November-2023])

Kaur, A. & Gulati, S., Samridhi & Singh. (2012). A comparative study of two formal specification languages: Z-notation and B-method. In *Proceedings of the second international conference on computational science, engineering and information technology* (pp. 524–531). Coimbatore UNK, India.

Kayan, H., Nunes, M., Rana, O., Burnap, P. & Perera, C. (2021). Cybersecurity of industrial cyber-physical systems: a review. *ACM Computing Surveys (CSUR).*

Kemp, C., Calvert, C. & Khoshgoftaar, T. (2018). Utilizing netflow data to detect slow read attacks. In *2018 ieee international conference on information reuse and integration (iri)* (pp. 108–116).

Kemp, C., Calvert, C. & Khoshgoftaar, T. M. (2020a). Detection methods of slow read dos using full packet capture data. In *2020 ieee 21st international conference on information reuse and integration for data science (iri)* (pp. 9–16).

Kemp, C., Calvert, C. & Khoshgoftaar, T. M. (2020b). Netflow feature evaluation for the detection of slow read http attacks. In *Reuse in intelligent systems*

(pp. 181–219). CRC Press.

Kemp, C., Calvert, C., Khoshgoftaar, T. M. & Leevy, J. L. (2023). An approach to application-layer dos detection. *Journal of Big Data*, *10*(1), 22.

Keshav, S. (2007, July). How to read a paper. *SIGCOMM Computer Communication Review*, *37*(3), 83—84. doi: https://doi.org/10.1145/1273445.1273458

Khan, M. U., Sherin, S., Iqbal, M. Z. & Zahid, R. (2019). Landscaping systematic mapping studies in software engineering: a tertiary study. *Journal of Systems and Software*, *149*, 396–436. doi: https://doi.org/10.1016/j.jss.2018.12.018

Khraisat, A., Gondal, I., Vamplew, P. & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1–22.

Kim, J., Chon, S. & Park, J. (2019). Suggestion of testing method for industrial level cyber-physical system in complex environment. In *2019 ieee international conference on software testing, verification and validation workshops (icstw)* (pp. 148–152). Xian, China.

kingsland, u. (n.d.). *6 Reasons Why IT Systems Administrators Need Cybersecurity Training.* `https://kingslanduniversity.com/systems-administrators-need-training`. ([Online; accessed 11-Novemeber-2023])

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M. & Linkman, S. (2010). Systematic literature reviews in software engineering – a tertiary study. *Information and Software Technology*, *52*(8), 792–805. doi: https://doi.org/10.1016/j.infsof.2010.03.006

Knüppel, A., Jatzkowski, I., Nolte, M., Thüm, T., Runge, T. & Schaefer, I. (2020). Skill-based verification of cyber-physical systems. In *International conference on fundamental approaches to software engineering. lecture notes in computer science* (Vol. 12076, pp. 203–223). doi: https://doi.org/10.1007/978-3-030-45234-6_10

Kordestani, M., Chaibakhsh, A. & Saif, M. (2020). Sms–a security management system for steam turbines using a multisensor array. *IEEE Systems Journal*, *14*(3), 3813–3824.

Kordestani, M. & Saif, M. (2021). Observer-based attack detection and mitigation for cyberphysical systems: A review. *IEEE Systems, Man, and Cybernetics Magazine*, *7*(2), 35–60.

Krotofil, M., Cárdenas, A. A., Manning, B. & Larsen, J. (2014). Cps: Driving cyber-physical systems to unsafe operating conditions by timing dos attacks on sensor signals. In *Proceedings of the 30th annual computer security applications conference* (pp. 146–155).

Krueger, M., Walden, D. & Hamelin, R. (2011). *Systems engineering handbook: a guide for system life cycle processes and activities (v. 3.2. 1).* International Council on Systems Engineering (INCOSE), San Diego, CA.

Kulvatunyou, B., Wallace, E., Ivezic, N. & Lee, Y. (2014, 09). Toward manufacturing system composability analysis: a use case scenario. In *Advances in production management systems* (Vol. 439, pp. 658–666). Ajaccio, France: Springer. doi: https://doi.org/10.1007/978-3-662-44736-9_80

Kumar, P., Goswami, D., Chakraborty, S., Annaswamy, A., Lampka, K. & Thiele, L. (2012). A hybrid approach to cyber-physical systems verification. In *Dac '12: The 49th annual design automation conference 2012* (pp. 688–696). San Francisco, California.

Kupwade Patil, H. & Chen, T. M. (2013). Wireless sensor network security. In J. R. Vacca (Ed.), *Computer and information security handbook (second edition)* (p. 301-322). Boston: Morgan Kaufmann. doi: https://doi.org/10.1016/B978-0-12-394397-2.00016-7

Lan, J., Lu, J. Z., Wan, G. G., Wang, Y. Y., Huang, C. Y., Zhang, S. B., ... Ma, J. N. (2022). E-minbatch graphsage: An industrial internet attack detection model. *Security and Communication Networks*, *2022*.

Lana, C. A., Guessi, M., Antonino, P. O., Rombach, D. & Nakagawa, E. Y. (2019). A systematic identification of formal and semi-formal languages and techniques for software-intensive systems-of-systems requirements modeling. *IEEE Systems Journal*, *13*(3), 2201–2212. doi: https://doi.org/10.1109/JSYST.2018.2874061

Langley, A., Riddoch, A., Wilk, A., Vicente, A., Krasic, C., Zhang, D., ... others (2017). The quic transport protocol: Design and internet-scale deployment. In *Conf. of the acm special interest group on data communication* (pp. 183–196).

Lee, E. (2008). Cyber physical systems: design challenges. In *Proceedings of the 11th ieee international symposium on object/component/service-oriented real-time distributed computing* (pp. 440–451).

Lee, J., Bagheri, B. & Kao, H.-A. (2015). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing letters*, *3*, 18–23.

Leitão, P., Barbosa, J., Funchal, G. & Melo, V. (2020). Self-organized Cyber-Physical Conveyor System using Multi-agent Systems. *International Journal of Artificial Intelligence*, *18*(2), 171–185. Retrieved from `http://www.ceser.in/ceserp/index.php/ijai/article/view/6578`

Leitão, P., Colombo, A. & Karnouskos, S. (2015, 09). Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry*, *81*. doi: 10.1016/j.compind.2015.08.004

LeMay, E., Ford, M. D., Keefe, K., Sanders, W. H. & Muehrcke, C. (2011). Model-based security metrics using adversary view security evaluation (advise). In *8th international conference on quantitative evaluation of systems(qest)* (pp. 191–200). Aachen, Germany.

Leng, Q., Qi, H., Miao, J., Zhu, W., Su, G. et al. (2015). One-class classification with extreme learning machine. *Mathematical problems in engineering*, *2015*.

Li, B., Wang, Y., Xu, K., Cheng, L. & Qin, Z. (2022). Dfaid: Density-aware and feature-deviated active intrusion detection over network traffic streams. *Computers & Security*, *118*, 102719.

Li, B., Wu, Y., Song, J., Lu, R., Li, T. & Zhao, L. (2021). DeepFed: Federated Deep

Learning for Intrusion Detection in Industrial Cyber–Physical Systems. *IEEE Transactions on Industrial Informatics*, *17*(8), 5615-5624. doi: 10.1109/TII.2020.3023430

Li, F., Zhang, P., Huang, H. & Chen, G. (2016, 03). A model-based service-oriented integration strategy for industrial CPS. In *International conference on industrial iot technologies and applications* (Vol. 173, pp. 222–230). GuangZhou, China. doi: https://doi.org/10.1007/978-3-319-44350-8_22

Li, N., Tsigkanos, C., Jin, Z., Hu, Z. & Ghezzi, C. (2020). Early validation of cyber–physical space systems via multi-concerns integration. *Journal of Systems and Software*, *170*, 110–742.

Li, Y., Qiu, R. & Jing, S. (2018). Intrusion detection system using online sequence extreme learning machine (os-elm) in advanced metering infrastructure of smart grid. *PloS one*, *13*(2), e0192216.

Liang, N.-Y., Huang, G.-B., Saratchandran, P. & Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on neural networks*, *17*(6), 1411–1423.

Lima, A. R., Hsieh, W. W. & Cannon, A. J. (2017). Variable complexity online sequential extreme learning machine, with applications to streamflow prediction. *Journal of Hydrology*, *555*, 983–994.

Lima, B. & Faria, J. P. (2018). Towards real-time patient prioritization in hospital emergency services. In *Ieee 20th international conference on e-health networking, applications and services (healthcom)* (pp. 1–4). Ostrava, Czech Republic.

Lima Filho, F. S. d., Silveira, F. A., de Medeiros Brito Junior, A., Vargas-Solar, G. & Silveira, L. F. (2019). Smart detection: an online approach for dos/ddos attack detection using machine learning. *Security and Communication Networks*, *2019*, 1–15.

Lin, Q., Adepu, S., Verwer, S. & Mathur, A. (2018). Tabor: a graphical model-based approach for anomaly detection in industrial control systems. In *Proceedings of the 2018 on asia conference on computer and communications security* (pp. 525–536). Incheon Republic of Korea. doi: https://doi.org/10.1145/3196494.3196546

Liu, B., Chen, J. & Hu, Y. (2022). Mode division-based anomaly detection against integrity and availability attacks in industrial cyber-physical systems. *Computers in Industry*, *137*, 103609.

Liu, F., Zhang, S., Ma, W. & Qu, J. (2022). Research on attack detection of cyber physical systems based on improved support vector machine. *Mathematics*, *10*(15). doi: 10.3390/math10152713

Liu, J., Zhang, W., Ma, T., Tang, Z., Xie, Y., Gui, W. & Niyoyita, J. P. (2020). Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection. *Expert Systems with Applications*, *158*, 113578.

Liu, L. & Li, Z. (2010). Improving parallelism and locality with asynchronous algorithms. *ACM Sigplan Notices*, *45*(5), 213–222.

Loucopoulos, P. & Karakostas, V. (1995). *System requirements engineering*. McGraw-Hill, Inc.

Loucopoulos, P., Kavakli, E. & Chechina, N. (2019). Requirements engineering for cyber physical production systems. In *31st international conference on advanced information systems engineering* (pp. 276–291). Rome, Italy.

Lu, A.-Y. & Yang, G.-H. (2017). Input-to-state stabilizing control for cyber-physical systems with multiple transmission channels under denial of service. *IEEE Transactions on Automatic Control*, *63*(6), 1813–1820.

Lu, Y., Morris, K. C. & Frechette, S. (2016). Current standards landscape for smart manufacturing systems. *National Institute of Standards and Technology, NISTIR*, *8107*, 39.

Ma, X., Almutairi, L., Alwakeel, A. M. & Alhameed, M. H. (2023). Cyber Physical System for Distributed Network Using DoS Based Hierarchical Bayesian Network. *Journal of Grid Computing*, *21*(2), 27.

Mahjabin, T., Xiao, Y., Sun, G. & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, *13*(12), 1550147717741463.

Mahmoud, M. S., Hamdan, M. M. & Baroudi, U. A. (2020). Secure control of cyber physical systems subject to stochastic distributed dos and deception attacks. *International Journal of Systems Science*, *51*(9), 1653–1668.

Mancini, T., Mari, F., Melatti, I., Salvo, I., Gruber, J., Hayes, B., ... Elmegaard, L. (2018, 10). Parallel statistical model checking for safety verification in smart grids. In *Ieee international conference on smart grid communications (smartgridcomm)* (pp. 1–6). Aalborg, Denmark. doi: https://doi.org/10.1109/SmartGridComm.2018.8587416

Mandayam K., S. & Steven P., M. (1995). *Formal verification of an avionics microprocessor* (Tech. Rep.). CSL-95-04: Technical report, SRI International Computer Science Laboratory.

Mann, C. (2009). A practical guide to sysml: the systems modeling language. *Kybernetes*, *38*. doi: https://doi.org/10.1108/k.2009.06738aae.004

Marin, G. A. (2005). Network security basics. *IEEE security & privacy*, *3*(6), 68–72.

MARK, B. (2003). *Demystifying mixed signal test methods*. Burlington: Newnes.

Mashkoor, A. & Hasan, O. (2012). Formal probabilistic analysis of cyber-physical transportation systems. In *International conference on computational science and its applications* (Vol. 7335, pp. 419–434). Salvador de Bahia, Brazil,. doi: https://doi.org/10.1007/978-3-642-31137-6_32

McCune, J. M., Shi, E., Perrig, A. & Reiter, M. K. (2005). Detection of denial-of-message attacks on sensor network broadcasts. In *Ieee symposium on security and privacy (s&p'05)* (pp. 64–78).

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D. & Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12-22. doi: 10.1109/MPRV.2018.03367731

Menghi, C., Nejati, S., Gaaloul, K. & Briand, L. C. (2019). Generating automated and online test oracles for simulink models with continuous and uncertain behaviors. In *Proceedings of the 2019 27th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering* (p. 27—38). Tallinn, Estonia. doi: https://doi.org/10.1145/3338906.3338920

Meseguer, J. & Ölveczky, P. (2012, 01). Formalization and correctness of the pals architectural pattern for distributed real-time systems. *Theoretical Computer Science*, *451*, 1–37. doi: https://doi.org/10.1016/j.tcs.2012.05.040

Metsälä, S., Gulzar, K., Vyatkin, V., Gröhn, L., Väänänen, E., Saikko, L. & Nyholm, M. (2017). Simulation-enhanced development of industrial cyber-physical systems using OPC-UA and IEC 61499. In *International conference on industrial applications of holonic and multi-agent systems* (pp. 125–139). Lyon, France.

Michael, T., Atif, M., Andreas, D. & Alexander, E. (2020). Ensuring safe and consistent coengineering of cyber physical production systems: a case study. *Journal of Software Evolution and Press*, *32*(2). doi: https://doi.org/10.1002/smr.2308

Micro, T. (2019). *The IIoT Attack Surface: Threats and Security Solutions.* Retrieved 2023-01-15, from `https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/-the-iiot-attack-surface-threats-and-security-solutions`

Misson, H. A., Gonçalves, F. S. & Becker, L. B. (2019). Applying integrated formal methods on CPS design. In *Ix brazilian symposium on computing systems engineering (sbesc)* (pp. 1–8). Natal, Brazil. doi: https://doi.org/10.1109/SBESC49506.2019.9046084

Mittal, M., Kumar, K. & Behal, S. (2022). Deep learning approaches for detecting ddos attacks: A systematic review. *Soft Computing*, 1–37.

Mizrahi, T. (2014, 8). *Security requirements of time protocols in packet switched networks* (RFC No. 7384). RFC Editor. Internet Requests for Comments. Retrieved from `https://datatracker.ietf.org/doc/html/rfc7384#section-3.2.4)`

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., ... Ueda, K. (2016). Cyber-physical systems in manufacturing. *Cirp Annals*, *65*(2), 621–641. doi: https://doi.org/10.1016/j.cirp.2016.06.005

Moustafa, N. & Slay, J. (2015). Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (milcis)* (p. 1-6). doi: 10.1109/MilCIS.2015.7348942

Moustafa, N. & Slay, J. (2018). A network forensic scheme using correntropy-variation for attack detection. In *Advances in digital forensics xiv: 14th ifip wg 11.9 international conference, new delhi, india, january 3-5, 2018, revised selected papers 14* (pp. 225–239).

Mühlfelder, M. (2018). Analysis and design of a cyber-physical production system

(CPPS) in sensor manufacturing. A case study. In *Proceedings of the 20th congress of the international ergonomics association (iea 2018)* (Vol. 822, pp. 391–400). Florence, Italy: Springer. doi: https://doi.org/10.1007/978-3-319-96077-7_41

Muntean, M. & Militaru, F. D. (2022). Design science research framework for performance analysis using machine learning techniques. *Electronics*, *11*(16), 2504.

Munz, G. & Carle, G. (2007). Real-time analysis of flow data for network attack detection. In *2007 10th ifip/ieee international symposium on integrated network management* (p. 100-108). doi: 10.1109/INM.2007.374774

Muralee-dharan, N. & Janet, B. (2021). A deep learning based http slow dos classification approach using flow data. *ICT Express*, *7*(2), 210–214.

Murvay, P.-S. & Groza, B. (2018). A brief look at the security of devicenet communication in industrial control systems. In *Proceedings of the central european cybersecurity conference 2018* (pp. 1–6).

Nägele, T., Broenink, T., Hooman, J. & Broenink, J. (2019). Early analysis of cyber-physical systems using co-simulation and multi-level modelling. In *2019 ieee international conference on industrial cyber physical systems (icps)* (pp. 133–138). Taipei, Taiwan.

Neghina, M., Zamfirescu, C.-B. & Pierce, K. (2019). Early-stage analysis of cyber-physical production systems through collaborative modelling. *Software and Systems Modeling*, 1–20.

Nejati, S., Gaaloul, K., Menghi, C., Briand, L. C., Foster, S. & Wolfe, D. (2019). Evaluating model testing and model checking for finding requirements violations in simulink models. In *Proceedings of the 2019 27th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering* (p. 1015—1025). Tallinn, Estonia. doi: https://doi.org/10.1145/3338906.3340444

Nizam, F., Chaki, S., Al Mamun, S., Kaiser, M. S. et al. (2016). Attack detection and prevention in the cyber physical system. In *2016 international conference on computer communication and informatics (iccci)* (pp. 1–6).

Nooribakhsh, M. & Mollamotalebi, M. (2020). A review on statistical approaches for anomaly detection in ddos attacks. *Information Security Journal: A Global Perspective*, *29*(3), 118–133.

Nuzzo, P., Li, J., Sangiovanni-Vincentelli, A. L., Xi, Y. & Li, D. (2019). Stochastic assume-guarantee contracts for cyber-physical system design. *ACM Transactions on Embedded Computing Systems (TECS)*, *18*(1), 1–26.

Nuzzo, P., Lora, M., Feldman, Y. A. & Sangiovanni-Vincentelli, A. L. (2018). CHASE: contract-based requirement engineering for cyber-physical system design. In *2018 design, automation & test in europe conference & exhibition (date)* (pp. 839–844). Dresden, Germany.

Object Management Group. (2011). *UML profile for modeling and analysis of real-time and embedded systems (MARTE)*. Object Management Group.

Offermann, P., Levina, O., Schönherr, M. & Bub, U. (2009). Outline of a design

science research process. In *Proceedings of the 4th international conference on design science research in information systems and technology* (pp. 1–11). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1555619.1555629

Ölveczky, P. C. & Meseguer, J. (2007). Semantics and pragmatics of real-time maude. *Higher-order and symbolic computation*, *20*(1-2), 161–196.

Omer, Y. (n.d.). *DDoS Attack Trends for Q1 2023.* `https://radar.cloudflare.com/reports/ddos-2023-q1`. ([Online; accessed 25-August-2023])

Omer, Y. & Pacheco, J. (2023). *DDoS threat report for 2023 Q3.* `https://blog.cloudflare.com/ddos-threat-report-2023-q3/`. ([Online; accessed 5-Novemeber-2023])

Omer, Y. & Vivik, G. (2021). *DDoS Attack Trends for Q4 2021.* `https://blog.cloudflare.com/ddos-attack-trends-for-2021-q4/`. ([Online; accessed 15-February-2022])

Onwubiko, C. (2020). Focusing on the recovery aspects of cyber resilience. In *2020 international conference on cyber situational awareness, data analytics and assessment (cybersa)* (pp. 1–13).

Oppenheim, A. V. & Schafer, R. W. (1998). *Discrete-time signal processing* (2nd ed.). USA: Prentice Hall Press.

Ortega-Fernandez, I. & Liberati, F. (2023). A review of denial of service attack and mitigation in the smart grid using reinforcement learning. *Energies*, *16*(2), 635.

Öztemel, E. & Gursev, S. (2020). Literature review of industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*, 127–182. doi: https://doi.org/10.1007/S10845-018-1433-8

Pagliari, L., Mirandola, R. & Trubiani, C. (2019, 07). Engineering cyber-physical systems through performance-based modelling and analysis: a case study experience report. *Journal of Software: Evolution and Process*, *32*(1). doi: https://doi.org/10.1002/smr.2179

Patani, N. P. & Patel, R. (2017). A mechanism for prevention of flooding based ddos attack. *International Journal of Computational Intelligence Research*, *13*(1).

Penzenstadler, B. & Eckhardt, J. (2012). A requirements engineering content model for cyber-physical systems. In *2012 second ieee international workshop on requirements engineering for systems, services, and systems-of-systems (ress)* (pp. 20–29). Chicago, USA.

Petersen, K., Vakkalanka, S. & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: an update. *Information and Software Technology*, *64*, 1–18. doi: https://doi.org/10.1016/j.infsof.2015.03.007

Połap, D., Woźniak, M., Wei, W. & Damaševičius, R. (2018). Multi-threaded learning control mechanism for neural networks. *Future Generation Computer Systems*, *87*, 16–34.

Pratomo, B. A., Burnap, P. & Theodorakopoulos, G. (2018). Unsupervised

approach for detecting low rate attacks on network traffic with autoencoder. In *2018 international conference on cyber security and protection of digital services (cyber security)* (pp. 1–8).

Pretorius, B. & van Niekerk, B. (2020). Industrial internet of things security for the transportation infrastructure. *Journal of Information Warfare*, *19*(3), 50–67.

Qaiwmchi, N. A. H., Amintoosi, H. & Mohajerzadeh, A. (2020). Intrusion detection system based on gradient corrected online sequential extreme learning machine. *IEEE Access*, *9*, 4983–4999.

Quentin & Monnet. (2015). *Smac protocol vulnerabilities to dos attacks* (Tech. Rep.). Technical report, Creative common zeros 1 universal licence.

Raiyat Aliabadi, M., Seltzer, M., Vahidi-Asl, M. & Ghavamizadeh, R. (2021, 03). Artinali#: An efficient intrusion detection technique for resource-constrained cyber-physical systems. *International Journal of Critical Infrastructure Protection*, *33*, 100430. doi: 10.1016/j.ijcip.2021.100430

Rajagopalan, A., Jagga, M., Kumari, A. & Ali, S. T. (2017). A ddos prevention scheme for session resumption sea architecture in healthcare iot. In *2017 3rd international conference on computational intelligence & communication technology (cict)* (p. 1-5). doi: 10.1109/CIACT.2017.7977361

Rashid, A. & Hasan, O. (2020). Formal analysis of the continuous dynamics of cyber–physical systems using theorem proving. *Journal of Systems Architecture*, *112*. (accessed on 22 October 2020) doi: https://doi.org/10.1016/j.sysarc.2020.101850

Rashid, A., Siddique, U. & Tahar, S. (2019). Formal verification of cyber-physical systems using theorem proving. In *International workshop on formal techniques for safety-critical systems* (pp. 3–18). Shenzhen,China. doi: https://doi.org/10.1007/978-3-030-46902-3_1

Ravi, V., Chaganti, R. & Alazab, M. (2022). Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Computers and Electrical Engineering*, *102*, 108156.

Reed, A., Dooley, L. S. & Mostefaoui, S. K. (2021). A reliable real-time slow dos detection framework for resource-constrained iot networks. In *2021 ieee global communications conference (globecom)* (pp. 1–6).

Rescorla, E. (2022, 4). *Datagram transport layer security version 1.3* (RFC No. 9147). Internet Engineering Task Force (IETF). Retrieved from `https://datatracker.ietf.org/doc/html/rfc9147`

Ribeiro, F. G. C., Rettberg, A., Pereira, C. E. & Soares, M. S. (2016). An analysis of the value specification language applied to the requirements engineering process of cyber-physical systems. *IFAC-PapersOnLine*, *49*(30), 42–47. doi: https://doi.org/10.1016/j.ifacol.2016.11.123

Rios, V. D. M., Inácio, P. R., Magoni, D. & Freire, M. M. (2022). Detection and mitigation of low-rate denial-of-service attacks: A survey. *IEEE Access*, *10*, 76648–76668.

Rocchetto, M. & Tippenhauer, N. O. (2017). Towards formal security analysis of

industrial control systems. In *Proceedings of the 2017 acm on asia conference on computer and communications security* (pp. 114–126). Abu Dhabi, United Arab Emirates.

Roka, S. & Naik, S. (2017). Survey on signature based intruction detection system using multithreading. *Intl. J. Res. Granthaalayah*, *5*, 58–62.

Roscoe, B. (1998). *The theory and practice of concurrency*. Prentice Hall.

Rouzbahani, H. M., Bahrami, A. H. & Karimipour, H. (2021). A snapshot ensemble deep neural network model for attack detection in industrial internet of things. *AI-Enabled Threat Detection and Security Analysis for Industrial IoT*, 181–194.

Ruchkin, I. (2015). Towards integration of modeling methods for cyber-physical systems. In *The doctoral symposium at the 18th acm/ieee international conference of model-driven engineering languages and systems 2015 (models 2015)*. Ottawa, Canada.

Saghezchi, F. B., Mantas, G., Violas, M. A., de Oliveira Duarte, A. M. & Rodriguez, J. (2022). Machine learning for ddos attack detection in industry 4.0 cppss. *Electronics*, *11*(4), 602.

Sakarovitch, J. (2009). *Elements of automata theory*. USA: Cambridge University Press.

Saleh, M. S., Althaibani, A., Esa, Y., Mhandi, Y. & Mohamed, A. A. (2015). Impact of clustering microgrids on their stability and resilience during blackouts. In *International conference on smart grid and clean energy technologies (icsgce)* (pp. 195–200). Offenburg, Germany.

Salim, M. M., Rathore, S. & Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. *The Journal of Supercomputing*, *76*(7), 5320–5363.

Sambangi, S., Gondi, L. & Aljawarneh, S. (2022). A feature similarity machine learning model for ddos attack detection in modern network environments for industry 4.0. *Computers and Electrical Engineering*, *100*, 107955.

Sanford, F., Dov, D. & Yaniv, M. (2020). *Modeling Standards.* `https://www.sebokwiki.org/wiki/Modeling_Standards`. ([Online; accessed 27-October-2020])

Sanna Passino, F., Adams, N., Cohen, E., Evangelou, M. & Heard, N. A. (2023). Statistical cybersecurity: A brief discussion of challenges, data structures, and future directions.

Sanwal, M. U. & Hasan, O. (2013). Formal verification of cyber-physical systems: coping with continuous elements. In *International conference on computational science and its applications* (pp. 358–371). Ho Chi Minh, Vietnam.

Sarker, I. H., Furhad, M. H. & Nowrozy, R. (2021). Ai-driven cybersecurity: an overview, security intelligence modeling and research directions. *SN Computer Science*, *2*, 1–18.

Schneble, W. & Thamilarasu, G. (2019). Attack detection using federated learning in medical cyber-physical systems. In *2019 28th international conference on*

computer communication and networks, icccn (pp. 1–8).

Sean, N. (2022). *Understanding and stopping multi-vector ddos attacks.* `https://www.corero.com/blog/understanding-and-stopping-multi-vector-ddos-attacks/`. ([Online; accessed 25-February-2022])

Seceleanu, C. C., Johansson, M. E., Suryadevara, J., Sapienza, G., Seceleanu, T., Ellevseth, S. E. & Pettersson, P. (2017). Analyzing a wind turbine system: from simulation to formal verification. *Science of Computer Programming*, *133*, 216–242. doi: https://doi.org/10.1016/j.scico.2016.09.007

Sepúlveda, S., Cravero, A. & Cachero, C. (2016). Requirements modeling languages for software product lines: a systematic literature review. *Information and Software Technology*, *69*, 16–36.

Shafiq, U., Shahzad, M. K., Anwar, M., Shaheen, Q., Shiraz, M., Gani, A. et al. (2022). Transfer learning auto-encoder neural networks for anomaly detection of ddos generating iot devices. *Security and Communication Networks*, *2022*.

Shaik, T. A. & Kataoka, K. (2021). capsaeul: Slow http dos attack detection using autoencoders through unsupervised learning. In *Proceedings of the 16th asian internet engineering conference* (pp. 49–55).

Sharafaldin, I., Lashkari, A. H. & Ghorbani, A. A. (2018a). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, *1*, 108–116.

Sharafaldin, I., Lashkari, A. H. & Ghorbani, A. A. (2018b). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th international conference on information systems security and privacy.* SCITEPRESS. doi: 10.5220/0006639801080116

Sharafaldin, I., Lashkari, A. H., Hakak, S. & Ghorbani, A. A. (2019). Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 international carnahan conference on security technology (iccst)* (pp. 1–8).

Sharma, A., Rani, S., Shah, S. H., Sharma, R., Yu, F. & Hassan, M. M. (2023). An efficient hybrid deep learning model for denial of service detection in cyber physical systems. *IEEE Transactions on Network Science and Engineering*.

Sharma, A. & Singh, M. (2013). Comparison of the formal specification languages based upon various parameters. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *11*(5), 37–39.

Shimeng, W., Yuchen, J., Hao, L., Jiusi, Z., Shen, Y. & Okyay, K. (2022). An integrated data-driven scheme for the defense of typical cyber–physical attacks. *Reliability Engineering System Safety*, *220*, 108257. doi: https://doi.org/10.1016/j.ress.2021.108257

Simon, F., Felex, W. & Jivka, O. (2019). A guideline for the requirements engineering process of SMEs regarding to the development of CPS. In *2019 8th international conference on industrial technology and management (icitm)* (pp. 85–94). Cambridge, United Kingdom.

Simplilearn. (2023). *System Administrator Roles and Responsibilities | Skills.*

https://www.simplilearn.com/systems-administrator-article. ([Online; accessed 11-Novemeber-2023])

Singh, A. & Jain, A. (2018). Study of cyber attacks on cyber-physical system. In *Proceedings of 3rd international conference on internet of things and connected technologies (iciotct)* (pp. 26–27).

Singh, G. & Singh, B. (2017). Simple service discovery protocol based distributed reflective denial of service attack. *International Journal of Recent Trends in Engineering & Research*, *3*(12), 143–150.

Singh, J. & Behal, S. (2020). Detection and mitigation of ddos attacks in sdn: A comprehensive review, research challenges and future directions. *Computer Science Review*, *37*, 100279.

Singh, K., Dhindsa, K. S. & Bhushan, B. (2018). Threshold-based distributed ddos attack detection in isp networks. *Turkish Journal of Electrical Engineering and Computer Sciences*, *26*(4), 1796–1811.

Singh, N. K. & Wang, H. (2019). Virtual environment model of glucose homeostasis for diabetes patients. In *2019 ieee international conference on industrial cyber physical systems (icps)* (pp. 417–422). Taipei, Taiwan.

Sinha, R., Dowdeswell, B., Zhabelova, G. & Vyatkin, V. (2018). Torus: scalable requirements traceability for large-scale cyber-physical systems. *ACM Transactions on Cyber-Physical Systems*, *3*(2). doi: https://doi.org/10.1145/3203208

Sinha, R., Pang, C., Martínez, G. S., Kuronen, J. & Vyatkin, V. (2015). Requirements-aided automatic test case generation for industrial cyber-physical systems. In *20th international conference on engineering of complex computer systems (iceccs)* (pp. 198–201). Gold Coast, Australia.

Sivamohan, S., Sridhar, S. & Krishnaveni, S. (2023). Tea-ekho-ids: An intrusion detection system for industrial cps with trustworthy explainable ai and enhanced krill herd optimization. *Peer-to-Peer Networking and Applications*, 1–29.

Soheily-Khah, S., Marteau, P.-F. & Béchet, N. (2018). Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the iscx dataset. In *2018 1st international conference on data intelligence and security (icdis)* (pp. 219–226).

Somani, G., Gaur, M. S., Sanghi, D., Conti, M. & Buyya, R. (2017). Ddos attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, *107*, 30–48.

Sourbier, N., Desnos, K., Guyet, T., Majorczyk, F., Gesny, O. & Pelcat, M. (2022). Secure-gegelati always-on intrusion detection through gegelati lightweight tangled program graphs. *Journal of Signal Processing Systems*, *94*(7), 753–770.

Stanciu, A. (2017). Blockchain based distributed control system for edge computing. In *2017 21st international conference on control systems and computer science (cscs)* (pp. 667–671).

Stewart, R. (2007, 9). *Security attacks found against the stream control transmission protocol (sctp) and current countermeasures* (RFC No. 5062). Muenster Univ. of Applied Sciences. Internet Requests for Comments. Retrieved from `https://datatracker.ietf.org/doc/html/rfc5062`

Strauss, A. & Corbin, J. (1998). *Basics of qualitative research techniques*. Sage publications Thousand Oaks, CA.

Su, L. & Ye, D. (2018). A cooperative detection and compensation mechanism against denial-of-service attack for cyber-physical systems. *Information Sciences*, *444*, 122–134.

Sun, H., Liu, J., Chen, X. & Du, D. (2015). Specifying cyber physical system safety properties with metric temporal spatial logic. In *Asia-pacific software engineering conference (apsec)* (pp. 254–260). New Delhi, India.

Suvarna, M., Yap, K. S., Yang, W., Li, J., Ng, Y. T. & Wang, X. (2021). Cyber-physical production systems for data-driven, decentralized, and secure manufacturing—a perspective. *Engineering*.

Suwannalai, E. & Polprasert, C. (2020). Network intrusion detection systems using adversarial reinforcement learning with deep q-network. In *2020 18th international conference on ict and knowledge engineering (ict&ke)* (pp. 1–7).

Tahir, Z., Khan, A. Q. & Asad, M. (2019). Attack detection and identification in cyber physical systems: An example on three tank system. In *2019 15th international conference on emerging technologies (icet)* (pp. 1–6).

Takbiri, Y. & Amini, A. (2019, 11). A survey on large-scale requirements engineering. In *4th international conference on combinatorics, cryptography, computer science and computing*. Tehran, Iran.

Tan, S., Guerrero, J. M., Xie, P., Han, R. & Vasquez, J. C. (2020). Brief survey on attack detection methods for cyber-physical systems. *IEEE Systems Journal*, *14*(4), 5329-5339. doi: 10.1109/JSYST.2020.2991258

Tang, D., Zhang, S., Chen, J. & Wang, X. (2021). The detection of low-rate dos attacks using the sadbscan algorithm. *Information Sciences*, *565*, 229–247.

Taormina, R., Galelli, S., Tippenhauer, N. O., Salomons, E. & Ostfeld, A. (2017). Characterizing cyber-physical attacks on water distribution systems. *Journal of Water Resources Planning and Management*, *143*(5), 04017009.

TCPdump & LibPcap. (n.d.). `https://www.tcpdump.org/`. ([Online; accessed 4-June-2021])

Theelen, B., Florescu, O., Geilen, M., Huang, J., Putten, P. & Voeten, J. (2007, 05). Software/hardware engineering with the parallel object-oriented specification language. In *5th ieee/acm international conference on formal methods and models for codesign (memocode 2007)* (pp. 139–148). Washington, DC,United States. doi: https://doi.org/10.1109/MEMCOD.2007.371231

Tomar, A., Jeena, D., Mishra, P. & Bisht, R. (2020). Docker security: A threat model, attack taxonomy and real-time attack scenario of dos. In *2020 10th international conference on cloud computing, data science engineering (confluence)* (p. 150-155). doi: 10.1109/Confluence47617.2020.9058115

Tripathi, N. & Hubballi, N. (2021). Application layer denial-of-service attacks and defense mechanisms: a survey. *ACM Computing Surveys (CSUR)*, *54*(4), 1–33.

Tsobdjou, L. D., Pierre, S. & Quintero, A. (2022). An online entropy-based ddos flooding attack detection system with dynamic threshold. *IEEE Transactions on Network and Service Management*.

Uniting cyber security and machine learning: Advantages, challenges and future research. (2022). *ICT Express*, *8*(3), 313-321. doi: https://doi.org/10.1016/j.icte.2022.04.007

Utic, Z. & Ramachandran, K. (2022). A survey of reinforcement learning in intrusion detection. In *2022 1st international conference on ai in cybersecurity (icaic)* (pp. 1–8).

Vanney, I. (2021). *hping3 flood ddos.* `https://linuxhint.com/hping3/`. ([Online; accessed 21-April-2021])

Van-Roy, P. & Haridi, S. (2004). *Concepts, techniques, and models of computer programming*. MIT press.

Vedula, V., Lama, P., Boppana, R. V. & Trejo, L. A. (2021). On the detection of low-rate denial of service attacks at transport and application layers. *Electronics*, *10*(17), 2105.

Vegendla, A., Duc, A. N., Gao, S. & Sindre, G. (2018). A systematic mapping study on requirements engineering in software ecosystems. *Journal of Information Technology Research (JITR)*, *11*(1), 49–69.

Verma, P. & Bharot, N. (2023). A review on security trends and solutions against cyber threats in industry 4.0. In *2023 third international conference on secure cyber computing and communication (icsccc)* (pp. 397–402).

Verma, P., De Leon, M. P., Breslin, J. G. & O'Shea, D. (2023). Fedtiu: Securing virtualized plcs against ddos attacks using a federated learning enabled threat intelligence unit. In *2023 ieee international conference on smart computing (smartcomp)* (pp. 233–236).

Vogel-Heuser, B., Schütz, D., Frank, T. & Legat, C. (2014). Model-driven engineering of manufacturing automation software projects–A SysML-based approach. *Mechatronics*, *24*(7), 883–897.

Volarević, I., Tomić, M. & Milohanić, L. (2022). Network forensics. In *2022 45th jubilee international convention on information, communication and electronic technology (mipro)* (pp. 1025–1030).

von Birgelen, A. & Niggemann, O. (2018). Anomaly detection and localization for cyber-physical production systems with self-organizing maps. In *Improve-innovative modelling approaches for production systems to raise validatable efficiency* (Vol. 8, pp. 55–71). Springer Vieweg, Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-662-57805-6_4

Vuong, T. P., Loukas, G., Gan, D. & Bezemskij, A. (2015). Decision tree-based detection of denial of service and command injection attacks on robotic vehicles. In *2015 ieee international workshop on information forensics and security (wifs)* (pp. 1–6).

Wang, J. (2007). Petri nets for dynamic event-driven system modeling. *Handbook of Dynamic System Modeling*, *1*, 24.

Wang, J., Song, Y., Wu, X. & Dai, W. (2019). A semi-formal requirement modeling pattern for designing industrial cyber-physical systems. In *45th annual conference of the ieee industrial electronics society* (pp. 2883–2888). Lisbon, Portugal.

Wang, R., Song, X., Zhu, J. & Gu, M. (2011). Formal modeling and synthesis of programmable logic controllers. *Computers in Industry*, *62*(1), 23–31.

Wang, W., Harrou, F., Bouyeddou, B., Senouci, S.-M. & Sun, Y. (2022). Cyber-attacks detection in industrial systems using artificial intelligence-driven methods. *International Journal of Critical Infrastructure Protection*, *38*, 100542. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1874548222000300` doi: https://doi.org/10.1016/j.ijcip.2022.100542

Wang, W., Wang, Z., Zhou, Z., Deng, H., Zhao, W., Wang, C. & Guo, Y. (2021). Anomaly detection of industrial control systems based on transfer learning. *Tsinghua Science and Technology*, *26*(6), 821–832.

Wang, X., Tu, S., Zhao, W. & Shi, C. (2022). A novel energy-based online sequential extreme learning machine to detect anomalies over real-time data streams. *Neural Computing and Applications*, *34*(2), 823–831.

Wang, Z., Li, Z., He, D. & Chan, S. (2022). A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning. *Expert Systems with Applications*, *206*, 117671.

Wang, Z., Zeng, Y., Liu, Y. & Li, D. (2021). Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection. *IEEE Access*, *9*, 16062–16091.

Westman, J. & Nyberg, M. (2014). *Specifying and structuring requirements on cyber-physical systems using contracts* (Tech. Rep.). Machine Design (Dept.), Mechatronics.: KTH, School of Industrial Engineering and Management (ITM).

Wieringa, R., Maiden, N., Mead, N. & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, *11*(1), 102–107.

Wiesner, S., Hauge, J. B. & Thoben, K.-D. (2015). Challenges for requirements engineering of cyber-physical systems in distributed environments. In *Ifip international conference on advances in production management systems* (pp. 49–58). Tokyo, Japan.

Wiesner, S., Marilungo, E. & Thoben, K.-D. (2017). Cyber-physical product-service systems–challenges for requirements engineering. *International Journal of Automation Technology*, *11*(1), 17–28.

Wisniewski, R., Grobelna, I. & Karatkevich, A. (2020). Determinism in cyber-physical systems specified by interpreted petri nets. *Sensors*, *20*(19), 55–65. doi: https://doi.org/10.3390/s20195565

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (pp. 1–10).

Wortmann, A., Barais, O., Combemale, B. & Wimmer, M. (2019). Modeling languages in industry 4.0: an extended systematic mapping study. *Software and Systems Modeling*, 1–28.

Wu, M. & Moon, Y. B. (2017). Taxonomy of cross-domain attacks on cybermanufacturing system. *Procedia Computer Science*, *114*, 367–374.

Wu, M. & Moon, Y. B. (2018). Taxonomy for secure cybermanufacturing systems. In *Asme international mechanical engineering congress and exposition* (Vol. 52019, p. V002T02A067).

Wu, X., Goepp, V. & Siadat, A. (2020). Concept and engineering development of cyber physical production systems: a systematic literature review. *The International Journal of Advanced Manufacturing Technology*, 1–19.

Wu, Z., Yue, M., Li, D. & Xie, K. (2015). Sedp-based detection of low-rate dos attacks. *Int. J. Commun. Syst.*, *28*(11), 1772–1788. doi: 10.1002/dac.2783

Xiao, Y.-j., Xu, W.-y., Jia, Z.-h., Ma, Z.-r. & Qi, D.-l. (2017). Nipad: a non-invasive power-based anomaly detection scheme for programmable logic controllers. *Frontiers of Information Technology & Electronic Engineering*, *18*, 519–534.

Xu, B. & Zhang, L. (2013). Formal specification of cyber physical systems: three case studies based on clock theory. In *Ieee international conference on green computing and communications and ieee internet of things and ieee cyber, physical and social computing* (pp. 804–811). doi: https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.143

Xu, C., Shen, J. & Du, X. (2021). Low-rate dos attack detection method based on hybrid deep neural networks. *Journal of Information Security and Applications*, *60*, 102879.

Yaacoub, J.-P., Noura, M., Noura, H. N., Salman, O., Yaacoub, E., Couturier, R. & Chehab, A. (2020). Securing internet of medical things systems: Limitations, issues and recommendations. *Future Generation Computer Systems*, *105*, 581–606.

Yaacoub, J.-P. A., Salman, O., Noura, H. N., Kaaniche, N., Chehab, A. & Malli, M. (2020). Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and microsystems*, *77*, 103201.

Yadav, A. P. & Mishra, N. (2023). Privacy and security control approach for ddos attacks in cyber physical systems using deep learning. In *2023 2nd international conference for innovation in technology (inocon)* (pp. 1–7).

Yaltirakli, G. (2015). Slowloris. *github.com*. Retrieved from `https://github.com/gkbrk/slowloris`

Yan, Y., Tang, D., Zhan, S., Dai, R., Chen, J. & Zhu, N. (2019). Low-rate dos attack detection based on improved logistic regression. In *2019 ieee 21st international conference on high performance computing and communications; ieee 17th international conference on smart city; ieee 5th international*

conference on data science and systems (hpcc/smartcity/dss) (pp. 468–476).

Yasrab, R. (2018, 04). *Mitigating docker security issues.* (arXiv preprint, arXiv:1804.05039v1)

Ye-Jing, L., Ming-Cai, C., Guang-Quan, Z., Yu-zhen, S., Fei, F. & Xing-hua, H. (2013). A model for vehicular cyber-physical system based on extended hybrid automaton. In *8th international conference on computer science & education* (pp. 1305–1308). Colombo, Srilanka.

You, J., Li, J. & Xia, S. (2012). A survey on formal methods using in software development. In *Iet international conference on information science and control engineering 2012 (icisce 2012).* Shenzhen, China. doi: https://doi.org/10.1049/cp.2012.2353

Yu, W., Dillon, T., Mostafa, F., Rahayu, W. & Liu, Y. (2019). Implementation of industrial cyber physical system: challenges and solutions. In *Ieee international conference on industrial cyber physical systems (icps)* (pp. 173–178). Taipei, Taiwan.

Yuan, E. & Malek, S. (2012). A taxonomy and survey of self-protecting software systems. In *2012 7th international symposium on software engineering for adaptive and self-managing systems (seams)* (p. 109-118). doi: 10.1109/SEAMS.2012.6224397

Yue, T., Ali, S. & Zhang, M. (2015). RTCM: a natural language based, automated, and practical test case generation framework. In *Proceedings of the 2015 international symposium on software testing and analysis* (pp. 397–408). Baltimore MD, USA.

Zacchia Lun, Y., DÍnnocenzo, A., Smarra, F., Malavolta, I. & Di Benedetto, M. (2019). State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software*, *149*, 174-216. doi: https://doi.org/10.1016/j.jss.2018.12.006

Zahid, F., Funchal, G., Melo, V., Kuo, M. M., Leitao, P. & Sinha, R. (2022a). Ddos attacks on smart manufacturing systems: A cross-domain taxonomy and attack vectors. In *2022 ieee 20th international conference on industrial informatics (indin)* (pp. 214–219).

Zahid, F., Funchal, G., Melo, V., Kuo, M. M. Y., Leitao, P. & Sinha, R. (2022b). DDoS Attacks on Smart Manufacturing Systems: A Cross-Domain Taxonomy and Attack Vectors. In *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)* (p. 214-219). doi: 10.1109/INDIN51773.2022.9976172

Zahid, F., Kuo, M. M. & Sinha, R. (2021a). Light-Weight Active Security for Detecting DDoS Attacks in Containerised ICPS. In *18th international conf. on privacy, security and trust (pst)* (pp. 1–5).

Zahid, F., Kuo, M. M., Sinha, R., Funchal, G., Pedrosa, T. & "Leitão, P. (2024). Actively detecting multiscale flooding attacks & attack volumes in resource-constrained icps. *IEEE Transactions on Industrial Informatics*, 1-9. doi: 10.1109/TII.2024.3383520

Zahid, F., Kuo, M. M. Y. & Sinha, R. (2021b). Light-weight active security for

detecting ddos attacks in containerised icps. In *2021 18th international conference on privacy, security and trust (pst)* (p. 1-5). doi: 10.1109/PST52912.2021.9647782

Zahid, F., Tanveer, A., Kuo, M. Y. M. & Sinha, R. (2021). A systematic mapping of semi-formal and formal methods in requirements engineering of industrial cyber-physical systems. *J Intell Manuf (2021)*..

Zargar, S. T., Joshi, J. & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE Communications Surveys Tutorials*, *15*(4), 2046-2069. doi: 10.1109/SURV.2013.031413.00127

Zeb, K., Baig, O. & Asif, M. K. (2015). Ddos attacks and countermeasures in cyberspace. *2015 2nd World Symposium on Web Applications and Networking (WSWAN)*, 1-6.

Zhan, H., Lin, Q., Wang, S., Talpin, J.-P., Xu, X. & Zhan, N. (2019). Unified graphical co-modelling of cyber-physical systems Using AADL and Simulink/Stateflow. In *7th international symposium on unifying theories of programming 2019* (Vol. 11885, pp. 109–129). Porto, Portugal: Springer. doi: https://doi.org/10.1007/978-3-030-31038-7_6

Zhang, D., Tang, D., Tang, L., Dai, R., Chen, J. & Zhu, N. (2019). Pca-svm-based approach of detecting low-rate dos attack. In *2019 ieee 21st international conference on high performance computing and communications; ieee 17th international conference on smart city; ieee 5th international conference on data science and systems (hpcc/smartcity/dss)* (pp. 1163–1170).

Zhang, D., Wang, Q.-G., Feng, G., Shi, Y. & Vasilakos, A. (2021, 28 January). A survey on attack detection, estimation and control of industrial cyber–physical systems. *ISA Transactions*. doi: 10.1016/j.isatra.2021.01.036

Zhang, H. & Yu, T. (2020). Taxonomy of reinforcement learning algorithms. *Deep Reinforcement Learning: Fundamentals, Research and Applications*, 125–133.

Zhang, J., Pan, L., Han, Q.-L., Chen, C., Wen, S. & Xiang, Y. (2021). Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica*, *9*(3), 377–391.

Zhang, L. (2011, 08). Formal specification for real time cyber physical systems using aspect-oriented approach. In *Fifth international conference on theoretical aspects of software engineering* (pp. 213–216). doi: https://doi.org/10.1109/TASE.2011.37

Zhang, L. (2013a, 08). Aspect-oriented modeling for railway control systems. In *Ieee international conference on information and automation, icia 2013* (pp. 236–241). Yinchuan, China. doi: https://doi.org/10.1109/ICInfA.2013.6720302

Zhang, L. (2013b). Modeling railway cyber physical systems based on aadl. In *19th international conference on automation and computing* (pp. 1–6). London, United Kingdom.

Zhang, L. (2013c). Requirement analysis method for vehicular cyber physical

systems. In *Ieee 10th international conference on high performance computing and communications & 2013 ieee international conference on embedded and ubiquitous computing* (pp. 2096–2103). Okayama, Japan.

Zhang, L. (2013d). Requirement specification for transportation cyber physical systems. In *Ieee international conference on green computing and communications and ieee internet of things and ieee cyber, physical and social computing* (pp. 1486–1491). Beijing, China.

Zhang, L. (2013e). Specifying and modeling automotive cyber physical systems. In *Ieee 16th international conference on computational science and engineering* (pp. 603–610). Washington, DC, United States.

Zhang, L. (2014). Modeling large scale complex cyber physical control systems based on system of systems engineering approach. In *2014 20th international conference on automation and computing* (pp. 55–60). Cranfield, UK.

Zhang, L., He, J. & Yu, W. (2013). Test case generation from formal models of cyber physical system. *International Journal of Hybrid Information Technology*, *6*(3), 15–24.

Zhao, J., Shetty, S., Pan, J. W., Kamhoua, C. & Kwiat, K. (2019). Transfer learning for detecting unknown network attacks. *EURASIP Journal on Information Security*, *2019*, 1–13.

Zheng, X. & Julien, C. (2015). Verification and validation in cyber physical systems: research challenges and a way forward. In *2015 ieee/acm 1st international workshop on software engineering for smart cyber-physical systems* (pp. 15–18). Florence, Italy.

Zheng, X., Julien, C., Kim, M. & Khurshid, S. (2015). Perceptions on the state of the art in verification and validation in cyber-physical systems. *IEEE Systems Journal*, *11*(4), 2614–2627.

Zhijun, W., Wenjing, L., Liang, L. & Meng, Y. (2020). Low-rate dos attacks, detection, defense, and challenges: a survey. *IEEE access*, *8*, 43920–43943.

Appendix A

An Intrusion Detection System Dataset for a Multi-Agent Cyber-Physical Conveyor System (Manuscript 6)

A.1 Abstract

Industry 4.0 is built upon the foundation of connecting devices and systems via Internet of Things (IoT) technologies, with cyber-physical systems (CPS) serving as the backbone infrastructure. Although this approach brings numerous benefits like improved performance, responsiveness and reconfigurability, it also introduces security concerns, making devices and systems vulnerable to cyber attacks. There is a need for effective techniques to protect these systems, and the availability of datasets becomes essential to support the development of such techniques. This paper presents a dataset based on the collection of traffic information exchanged in a self-organizing conveyor system using the multi-agent

systems (MAS) architecture and containing various intelligent conveyor modules. The dataset comprises data collected at the network and agent levels under normal system operation, DoS, and malicious agent attacks. An intrusion detection system that integrates DFT and ML analysis is developed to demonstrate the utility of this dataset.

A.2 Introduction

Innovations in digital technologies are driving the fourth industrial revolution, also known as Industry 4.0, significantly upgrading production systems and business models. This revolution enables data-driven strategies that can help companies to become more responsive to strong global competition and dynamic customer demands. Industry 4.0 features the high device and system connectivity through the usage of Internet of Things (IoT) technologies, supplemented by other disruptive technologies such as Big Data, Cloud Computing, and Artificial Intelligence (AI).

CPS is a mainstay approach to develop smart Industry 4.0 compliant solutions (E. Lee, 2008; Leitão, Colombo & Karnouskos, 2015), based on a network of cyber and physical entities that are combined to achieve a certain goal, contributing to transform traditional factories into intelligent factories (Hellinger, Translation, Macfarlane, Services & Galloway, 2011). CPS is notably involved in the integration of sensors and actuators in order to achieve the system goals, and for that, IoT technologies are employed to facilitate the collaborative work in distributed systems (Chebudie, Minerva & Rotondi, 2015).

Industry 4.0 systems feature large exchanges of data between various devices, systems, and networks that can be used for analytics and decision-making, but can also create new opportunities for both industry and attackers. Benefits to industry

include enhanced productivity through optimization and automation, cost savings, real-time monitoring, better working conditions and sustainability, and improved agility, which can lead to increased efficiency, reduced downtime, and improved safety (Chowdhury & A Raut, 2019). On the other hand, the massive exchange of data also increases the attack surface for cyber attackers (Micro, 2019). The number of potential vulnerabilities that can be exploited increases proportionally to the number and diversity of interconnected devices and systems. Unfortunately, Industry 4.0 systems are high-value targets for attackers because successful attacks can cause catastrophic damage, such as disruption of critical infrastructure and significant financial and human costs.

To address cyber-security concerns in Industry 4.0, it is important to implement robust security measures. These measures typically include encryption, authentication, access control, and regular security assessments. Additionally, risk management is crucial; it helps in identifying and mitigating risks and minimizing the impact of a potential attack. In addition to traditional security measures, methods employing ML techniques have become increasingly popular. Consequently, the availability of robust and realistic cyber-security datasets directly related to Industry 4.0 is becoming increasingly important to train ML models ("Uniting cyber security and machine learning: Advantages, challenges and future research", 2022). Such datasets have some highly-desired characteristics:

- *Validity*: providing a more realistic representation of the CPS/IoT system can improve the validity of trained models.
- *Generalization*: providing insights and making accurate predictions about similar data or events that were not observed or included in the original dataset, covering a wide range of relevant features and variations that can be applied more broadly to similar cases.

- *Scalability*: capturing scaled-up scenarios as systems are expected to grow and handle increasing numbers of devices and data streams.
- *Robustness*: datasets capturing various types of noise and errors can improve the reliability of the trained models.
- *Real-world performance*: datasets based on measurements from real systems can help to identify potential impacts of trained models on overall system performance.

Having this in mind, this paper describes the creation of an intrusion detection system dataset based on the collection of the traffic information exchanged in a real self-organizing conveyor system that uses the multi-agent systems (MAS) technology. The exchanged data between the system's components is captured at the network and agents levels, bringing different perspectives to analyze the functioning of the system, during the normal operation and a number of attack scenarios. The use of the described dataset is illustrated to develop an effective intrusion detection system (IDS) based on the DFT analysis and ML techniques.

The rest of the paper is organized as follows: Section A.3 provides the related work and Section A.4 describes the development of the cyber-security dataset. Section A.5 presents the development of an IDS that consider the DFT and ML analysis and uses the proposed dataset. Section A.6 presents the conclusions of the work and points out the future work.

A.3 Related Work

The escalation of CPS attacks has attracted significant interest from the research community, either in detecting these attacks or even in implementing more advanced mechanisms to predict them. The need for creating realistic datasets to

assess the IDS performance in the CPS and IoT domains has been underlined in a number of recent research works.

Among some of the available datasets, the CIC IoT Dataset 2022 (Dadkhah et al., 2022) presents the collection of data extracted from various IoT devices to analyze the behavior displayed in different scenarios and situations, including major attacks in the IoT environment. The UNSW-NB15 dataset (Moustafa & Slay, 2015) presents a network intrusion detection dataset that contains 2.5 million network packets generated by a variety of network-based attacks (fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode and worms), as well as the normal network traffic used to evaluate network intrusion detection systems. Other well-known datasets are the N-BaIoT (Meidan et al., 2018), which contains a set of network-based attacks specifically targeting IoT devices, such as botnet attacks, and malware infections, and the CICIDS2017 (Sharafaldin, Lashkari & Ghorbani, 2018b), which contains a mix of benign and malicious network traffic, including a wide range of attack types such as DoS, DDoS and brute-force attacks.

These datasets can be used to train and evaluate IDS, to analyze trends in cyber incidents, and to identify common vulnerabilities and attack vectors. Although these datasets have provided valuable resources for evaluating IDS in CPS and IoT domains, they have some limitations. It is important to notice that datasets are usually collected over a period of time, so the threat landscape may have changed since the data was collected and new ones will always need to be created. In addition, only the statistical analysis of packets on the network may not be enough to detect some more elaborated attacks that are directly related to the system operation, e.g., an infiltrated malicious agent interacting with the system agents can interfere very efficiently in its operation without sending many packets. In this way, the capture of data by the agents (or even by internal system entities) brings another perspective of the system data, allowing more analysis to

be carried out.

A.4 Development of the Dataset

This section describes the development of the dataset based on a self-organized conveyor system that is regulated by using the MAS technology.

A.4.1 System Description

The system under study in this work consists of a modular and self-organized conveyor system that comprises a sequence of modular Fischetechnik conveyors, aiming to transport an object from a starting point to an ending point and using MAS to regulate its operation in a distributed manner. Each conveyor module is made up of a cyber-physical component, in which the physical part consists of a conveyor belt, a DC 24V motor and two photoelectric sensors used to detect objects in the input and output positions of the conveyor belt, and the cyber part comprises an agent running in a Raspberry Pi, as illustrated in Fig. A.1.

The agents were developed using the JADE (*JAVA Agent DEvelopment Framework*, n.d.) framework, being the operation of the entire system regulated by the exchange of messages between them over WiFi. The exchanged messages are formatted according to the FIPA Agent Communication Language (ACL). The transfer of the objects along the conveyor modules should obey to some precedences between them. As example, considering a pair of conveyor modules, the second conveyor starts its motor when a part arrives at the end of the first module, reaching its output sensor. In this situation, the agent running in the second conveyor system receives the "tokenTransmissionOutput" message sent by the agent of the previous conveyor. The first module stops its motor when the part arrives at the beginning of the second conveyor, being detected by its

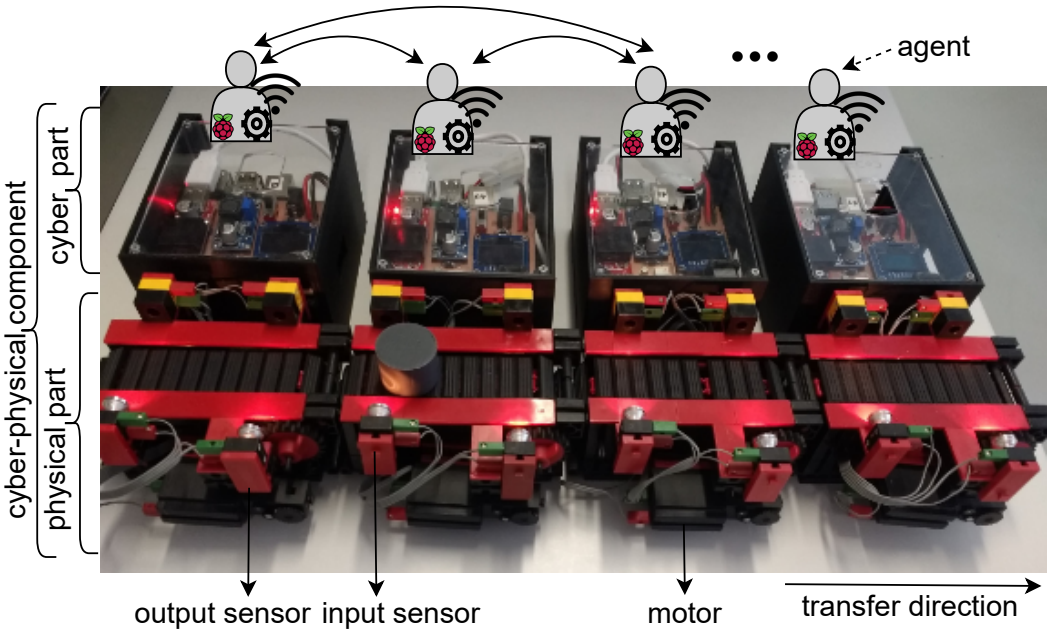


Figure A.1: MAS-based cyber-physical conveyor system.

input sensor. At this moment, the agent running in the first conveyor system receives the "tokenTransmissionInput" message sent by the agent of the posterior conveyor. More information about the operation of this system can be found in (Leitão et al., 2020).

A.4.2 Configuration of the Experimental Setup

In order to obtain only data from the system under study, an isolated network was used. For this purpose, a Cisco AIR-AP1121G-E-K9 router was used as an access point for all system components. Fig. A.2 presents the experimental setup, where each conveyor module has associated one agent running on a Raspberry Pi, and the JADE platform is running on another Raspberry Pi.

For the monitoring system, a machine with the Kali Linux operating system and an Alfa AWUS1900 network adapter was used, which aims to capture all messages exchanged on the network, being the collected data stored in the PotgreSQL database. Finally, the attacker/hacker is running in a machine with a Kali Linux

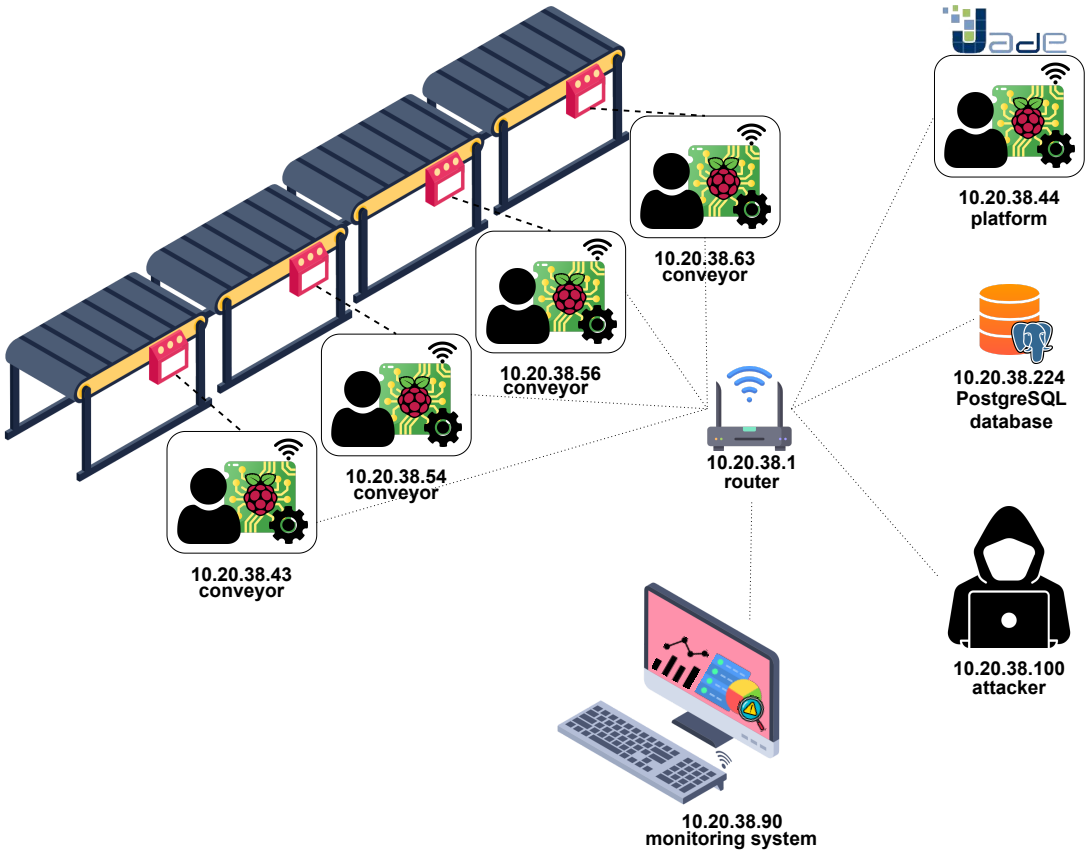


Figure A.2: Infrastructure setup to create the dataset.

operating system.

The interaction between the different devices presented in the experimental setup, illustrated in Fig. A.2, happens as follows: i) the agents of each module and the agents of the JADE platform exchange messages to keep the system functioning and carry out the intelligent and self-organizing tasks of the system, ii) all messages received by agents are sent to the database, iii) the monitoring system is responsible for capturing all exchanged messages on the wireless network, and iv) the attacker/hacker interacts with the system to interfere with its proper functioning, sometimes on the network, sometimes with the direct interaction with the agents.

A.4.3 Attack Generation

Some experiments were conducted to carry out some attacks on the modular conveyor system, with the corresponding network traffic being properly captured.

First, experiments aimed the DoS attack with the use of the `hping3` and `slowloris`(Yaltirakli, 2015) tools. `hping3` is a network tool able to send custom ICMP/UDP/TCP packets, allowing the creation of packets according to the desired criteria and sending them to the target IP. Slowloris is an HTTP DoS attack that affects threaded servers, being its operation consisting of making many HTTP requests, sending headers periodically to keep connections open, and creating a new session in case the server closes the connection to continue making requests. This causes the server's thread pool to be depleted and the server cannot respond to others.

In this way, `hping3` was used to send TCP packets to the JADE platform, which has the IP 10.20.38.44 and port 1099, in order to interfere in the management of the system in general. When carrying out the attacks, depending on the amount of packets sent per second, it could interrupt all system communication instantly and the data collection would be affected. Thus, it was decided to vary the amount of data sent per second to obtain more data collection, sending 10 and 100 packets per second. In the case of using Slowloris, 150 sessions were opened with the platform's IP (10.20.38.44) and data were collected. In both attacks, the delay generated in the exchange of messages between the agents was observed, which directly interfered with the functioning of the system, causing the transport not to be carried out in the desired way due to the agents not receiving messages.

On the other hand, at the agent level, a malicious agent was created to interact with the system and exploit the system vulnerabilities. The attack involves listening the exchanged messages, and whenever a message belonging to the

tokenTransmissionOutput protocol is received (which indicates that there is an object in the position of the output sensor of that module), the malicious agent copies this message with modified information. This causes agents to operate inappropriately, such as turning on the motors of various modules when there is no object to be delivered, which causes the motors of these modules to run unnecessarily.

A.4.4 Dataset Creation

For the creation of the dataset, after the configuration of the experimental setup, the system was put into operation and objects were transported to enable the communication between the different devices and the collection of exchanged data. The different operation modes that constitute the dataset are summarized in Table A.1.

The first experiment consisted of collecting data from the system running in its normal operation, which may have several modifications during operation, i.e modules were removed and added to the system, and the positions of the modules were also exchanged so that the reconfiguration was carried out. In this way, the normal operation of the system encompasses all possibilities for exchanging messages between agents. Another conducted experiment consists of sending TCP packets to the system's platform IP in order to flood it with incoming packets, at two different rates, i.e. 10 and 100 packets per second. While the packets were being sent, the system was put into operation and the objects were transported. However, as expected, there was a delay in the agents' reactions in turning on the conveyor motors, interfering with the system's operation and causing the modules (conveyors) to deny services (failing to transport objects correctly). The third experiment consisted of using the Slowloris tool to open several sessions with the

platform's IP and data were collected. Again, there was a delay in the agents' reaction. Finally, the last experiment considered the use of a malicious agent, as explained in the previous section, which listened all messages exchanged between agents, manipulated these messages and sent them again, completely interfering with the system's operation. In this case, every message sent by the attacker was identified in the collection, allowing to use supervised ML techniques for its posterior analysis.

A.4.5 Dataset Attributes

The dataset contains information collected from the agents and network level, being organized in 3 files for each previously referred collection, containing a `.cap` and a `.csv` file of the network collection and a `.csv` file of the agents collection, as illustrated in Table A.1. This dataset is available on the Polytechnic Institute of Bragança data repository website (Funchal et al., 2023).

The parameters of the data collected by the agents is described in Table A.2, and includes internal system information that brings a different perspective of the system operation. The data related to the malicious agent attack included 1.596 samples/messages sent and received by the agents, with 75% of the data belonging to the normal class and 25% of data to the attack class. This is due to the fact that the system exchanges a significant amount of information to maintain its functionality, even when it is being attacked by some external messages. For the other collections at the agent level, the amount of messages exchanged by the agents consist of true messages for the system, and there may be delays, and therefore, analyzes can be performed. On the other hand, regarding the data collected at the network level, the description of the attributes are presented in Table A.3.

Table A.1: Information about collected data.

Network and agent data collection	**Files extension**	**No. of packets**	**No. of messages exchanged**
Normal traffic	.cap & .csv	35.920	3.481
DoS attack (10 packets per second)	.cap & .csv	89.070	642
DoS attack (100 packets per second)	.cap & .csv	82.387	882
Slowloris DoS attack (150 sessions)	.cap & .csv	159.147	1.314
Malicious agent	.cap & .csv	17.957	1.596

A.5 Intrusion Detection using the Dataset

The usage of the described dataset is illustrated with the development of an IDS in a CPS that considers the proposed dataset, implemented through the use of two different approaches, namely FFT analysis and ML algorithms.

A.5.1 FFT Analysis

The effectiveness of the proposed dataset is determined by experimenting on the light-weight active security approach for DoS attack detection in a frequency domain (Zahid, Kuo & Sinha, 2021b). The approach has three major components, namely Pre-processing, Spectrum Calculation and Spectrum Analysis, and only considers the data related to the network traffic. During the pre-processing, discrete packets from the network traffic (in a time-domain) are distributed into contiguous and equally sized event bins. In the spectrum calculation, a light-weight Quick Discrete Fourier Transform (QuickDFT) approach is used to transform the packets in event bins into the frequency samples. The spectrum analysis comprises two-phases: attack presence and attack volume detection. To detect the presence of an attack in the traffic data, the first spectra similarity is performed by calculating the Jaccard Similarity between the spectra of normal traffic (set as a baseline) and the incoming traffic. Then, the modified fast-entropy method is used to determine the entropy (uncertainty) of the network traffic. On the basis of the

true output of the attack presence detection phase, the attack volume detection phase is triggered, which uses the Euclidean distance to find out the attack volume, i.e. one dominant peak (one-time extreme peak) or multiple dominant peaks (peak volume) for the attack traffic that is equal to or greater than the upper threshold (red line), as shown in Fig. A.3.

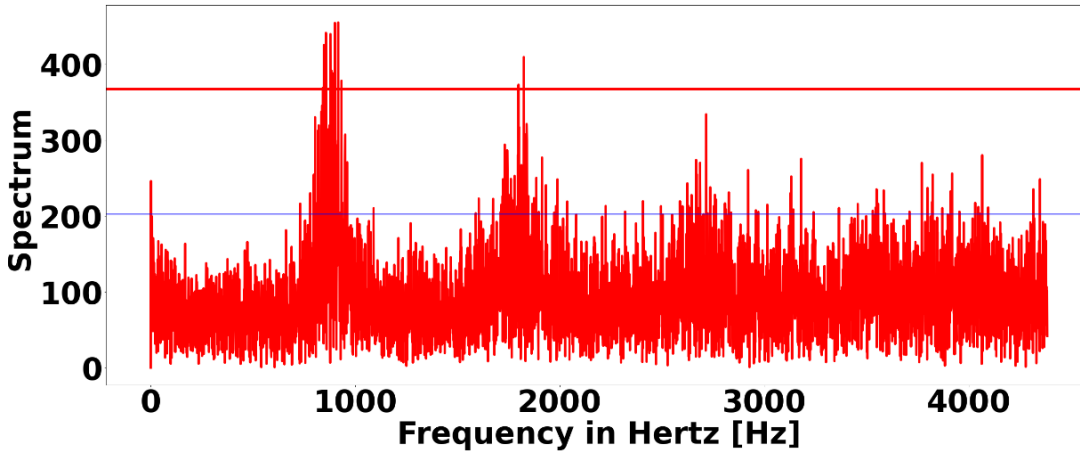


Figure A.3: Illustration of the attack traffic in a frequency domain with blue line (lower threshold) and red line (upper threshold).

To determine the performance of the proposed dataset on the DoS detection technique discussed above, the dataset was evaluated according to four performance evaluation criteria as illustrated in Table A.4: true positive rate, false positive rate, precision and accuracy. Briefly the *true positive rate* is a ratio of correctly identified attack samples to the total number of attack samples in the proposed dataset, being calculated by using Eq. **??**.

$$\text{true-positive rate} = \frac{TP}{TP + FN} \tag{A.1}$$

where the true positive (TP) means the correct identification of the DoS attack traffic and the false negative (FN) means the incorrect identification of DoS traffic as a normal traffic.

The *false positive rate* is the percentage of normal samples that are incorrectly identified as an attack sample and is computed by Eq. A.2.

$$\text{false-positive rate} = \frac{FP}{TN + FP} \tag{A.2}$$

where the false positive (FP) shows the incorrect identification of the normal traffic as DoS traffic and the true negative (TN) is the correct determination of normal traffic.

The *precision* determines how well a system can identify the attack or normal behavior and is calculated by Eq. A.3.

$$\text{precision} = \frac{TP}{TP + FP} \tag{A.3}$$

The *accuracy* is a measure to determine how correctly the technique detects an attacks and it is computed by Eq. A.4.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{A.4}$$

The results in Table A.4 show that the proposed dataset is effective to use in determining the DoS attacks, presenting an accuracy and precision equal to or greater than 96%, and a low FP rate (between 2% and 3%).

A.5.2 ML Analysis

In addition to apply the aforementioned DFT approach to the attack detection using the data captured on the network, the dataset was also experimented with the use of ML techniques to identify attacks at the agent level, i.e. the analysis of the messages exchanged by the agents.

The information extracted from the messages collected by the agents that were used in the application of an ML model contains the information presented in Table.A.2 with the exception of the ID, timestamp, sender, language, conversation_id and product_id. And since this is supervised data, the classification labels as 0 or 1 were used, being normal and attack messages, respectively.

The dataset related to the agent level included 1.596 samples/messages sent and received by the agents. Training and testing sets were created, with 1.117 data in the training set and 479 data in the testing set, representing 70% and 30% of the data, respectively. Several classification algorithms were examined in this work, namely Support Vector Machine (SVM), Random Forest (RF), Extra Trees Classifier (ETC), Decision Trees (DT), and Extreme Gradient Boosting (XGBoost).

The most well-known metrics were utilized to examine the success rate of applying these ML algorithms in the dataset, namely the *accuracy* (Eq. A.4), *precision* (Eq. A.3), *recall* (exactly the same of Eq. **??**), and *F1-score* that represents the harmonic mean of *precision* and *recall*, being calculated by using Eq. A.5.

$$\text{F1-score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{A.5}$$

The achieved results are summarised in Table A.5, being clear that all the algorithms exhibit remarkable results, evidenced by their high *precision* and *recall* values.

Nevertheless, the XGBoost algorithm emerged as the top performer. It can be seen that for the set of messages categorized as normal, 100% of them will be labeled correctly as normal, with 99% sensitivity meaning 1% of these messages may not actually be normal. As for attack messages, 97% of them will be accurately

labeled as attack, but 1% of these messages may mistakenly be categorized as normal. The *F1-score* values, which measure the average of *precision* and *recall*, are 99% and 98% for classes 0 (normal) and 1 (attack) respectively, indicating that on average, 99% of normal messages and 98% of attack messages will be accurately classified. The *accuracy* value, which reflects the correct classification rate, indicates that the application of the XGBoost algorithm to the test data set achieved a 99.19% accuracy rate.

A.6 Conclusions and Future Works

The article discusses the importance of addressing security issues in industrial CPS based on MAS. These systems are critical to ensuring the safety, efficiency, and reliability of industrial operations. However, as they become more complex and interconnected, they are also becoming more vulnerable to security threats. The development of a representative dataset of ICPS based on MAS can help identify vulnerabilities, prevent cyber attacks, and ensure the safety and efficiency of industrial operations. It was considered a case study related to a self-organized cyber-physical conveyor system, developed with MAS technology. The collection of data exchanged in the network and agents levels was carried out, allowing to use a set of distinct techniques to detect DoS attacks and also attacks by malicious agents in a MAS system.

In order to demonstrate the use of the dataset, two analysis techniques were applied, namely FFT and ML techniques. In the first case, the FFT technique was used to detect DoS attacks on the data captured in the network, performing the analysis in the frequency domain. On the other hand, some ML algorithms were applied to the data captured by the agents to detect attacks carried out by the malicious agent. Both techniques showed excellent results in the intrusion

detection, with an accuracy greater than 96%, showing the effectiveness and usefulness of the proposed dataset, that can be used to implement intrusion detection systems for this type of CPS.

Despite the fact that the attacks employed in this study may be relatively simple, they were designed to generate data that accurately reflects this type of system, in order to facilitate the development of more advanced intrusion detection systems using various techniques, e.g., ML. Future work will concentrate on enhancing these attacks with more complex scenarios, applying some tools in the collection of the network for the extraction of statistical features, and also expanding the data collection period to obtain greater resources for algorithm training.

Table A.2: Description of the dataset - agents level.

Attribute	**Description**
ID	ID to identify the sequence of message.
Timestamp	Unix time stamp, that allows to determine the time elapsed between each exchanged message.
Sender	Message sender.
Receiver	Message receiver.
Content	Content of the message sent by the agent.
Protocol	Protocol of the message sent by the agent. Agents communicate by exchanging messages using specific protocols that dictate how information is transmitted and interpreted. The protocols used depend on the design and functioning of the agents involved, and allow them to understand the meaning of the messages and respond appropriately. The use of consistent protocols is essential to enable effective communication and collaboration among agents.
Language	Language of the message sent by the agent, most often FIPA-ACL.
Conversation_id	A conversation_id is a unique identifier assigned to a message exchange between agents. This ID allows tracking and identification of the complete conversation between them. If an agent receives a message and provides a response, that response will have the same conversation_id as the original message. It ensures effective communication and coordination among agents.
Product_id	A product_id is a unique identifier that represents a specific part being transported. When messages are exchanged between agents and contain the same product _id, they refer to messages related to the transport of that specific part.
Conveyor_id	A conveyor_id is a unique identifier that represents an agent's current position in a transport sequence, e.g., if an agent is in the second position to perform the transport, its conveyor_id would be equal to 2.
Processing_time	Processing_time refers to the total time taken to receive and process a message. This time is calculated by taking the difference between the timestamp associated with the sending of the message and the timestamp of its receipt.
N_conveyors	Number of active agents/conveyors in the system, which allows to analyze the amount of messages exchanged in relation to the number of agents in the system and identify whether it follows a pattern or an average of messages.
Amount_messages	Number of messages that were exchanged before receiving that specific message. This is reset at each insertion of a new part to be transported. In this way, it is possible to analyze how many messages are being exchanged for the transport of each part.

Table A.3: Description of the dataset - network level

Attribute	**Description**
Number	Frame number from the beginning of the pcap.
Time	Seconds broken down to the nanosecond from the first frame of the pcap.
Delta	Delta time displayed.
Source	Source address (IPv4).
Destination	Destination address (IPv4).
Protocol	Protocol used in the Ethernet frame, IP packet, or TCP segment.
Length	Length of the frame in bytes.
Info	Information/details about the packet.

Table A.4: Results for the different attack rates for the FFT analysis.

Attack Period	**TP Rate**	**FP Rate**	**Precision**	**Accuracy**
10 packets/sec	97%	2%	98%	97.5%
100 packets/sec	99%	3%	96%	96%

Table A.5: Results for the analysis of the ML techniques

Algorithm	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**
SVM	0	0.98	0.98	0.98	96.96%
	1	0.93	0.92	0.93	
RF	0	1.00	0.99	0.99	98.78%
	1	0.95	0.99	0.97	
ETC	0	1.00	0.98	0.99	98.18%
	1	0.93	0.99	0.96	
DT	0	1.00	0.99	0.99	98.78%
	1	0.95	0.99	0.97	
XGBoost	0	1.00	0.99	0.99	99.19%
	1	0.97	0.99	0.98	