DOMAIN ADAPTATION OF DEEP LEARNING (D)DOS ATTACK DETECTION MODELS IN RESOURCE-CONSTRAINED CYBER PHYSICAL SYSTEMS ENVIRONMENTS

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF PHILOSOPHY

Supervisors
Dr. Roopak Sinha
Dr. Kirill Levchenko
Dr. Mahsa Mohaghegh

June 2023

By
Vicky NGO
School of Engineering, Computer and Mathematical Sciences

Abstract

Cyber-Physical systems (CPS) can broadly be defined as the integration of communication, control, and software components into physical processes. When such a system applies to industrial process control, this system is referred to as an industrial control system (ICS), whose purpose is to monitor and control physical industrial processes. Due to the high availability requirements present in Industrial Control Systems (ICS), any cyberattacks that can interrupt its processes are unacceptable. (Distributed) denial-of-service ((D)DoS) attacks are examples of such attacks.

With the advancement of cyber-integration and network communication in ICS and CPS, investment is needed to protect systems against (D)DoS. In recent years, there has been research on using machine learning and deep learning algorithms for (D)DoS attacks in ICS, as well as in CPS and IoT. However, existing studies do not sufficiently address the different existing types of (D)DoS attacks while also maintaining low computational overhead in resource-constrained environments.

This research investigates the adaptability and flexibility of existing detection algorithms for different attack types in multiple domains, particularly ICS, IoT, and CPS. Our hypothesis is that it is theoretically possible to adapt a detection model to the CPS and IoT domain, and vice versa, based on the datasets it trained on within some constraints.

Using a controlled experiment research methodology, we trained each of the three different detection models on three datasets: CICIDS2017, CICIDDoS2019, and the SWaT. The models were then evaluated on a Raspberry PI to measure their computational overhead. We found that a model's capability for domain adaptation is largely dependent on the model's architecture. Particularly, the model's architecture must be sufficiently flexible to extract and learn from relevant features in an unfamiliar detection domain. Additionally, we also identify various impacts that domain adaptation might have on a model, which include detection performance and computational overhead. This inherently affects the model's applicability for deployment into a resource-constrained system in the real world.

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 student

Contribution to Co-Authored Works

All co-authors on chapters/papers indicated in the following table have approved these for inclusion in this thesis.

Publication	**Contribution**
Ngo, V., Zahid, F., Mohaghegh, M., Sinha, R. (2022) A Systematic Mapping of Datasets and Machine Learning Models for (D)DoS Detection in Industrial Control System. IEEE Transactions on Neural Networks and Learning Systems, Manuscript ID: TNNLS-2022-S-25470 *(Chapter 2 and Chapter 3)* **Under review**	Vicky Ngo = 85% Farzana Zahid = 5% Mahsa Mohaghegh = 5% Roopak Sinha = 5%
Ngo, V., Mohaghegh, M., Sinha, R. (2022) Domain Adaptation of Deep Learning (D)DoS Attack Detection Models for Cyber-Physical Systems and Internet-of-Things. IEEE Transactions on Artificial Intelligence, Manuscript ID: [TAI-2023-May-A-00425] *(Chapter 5 and Chapter 6)* **Under review**	Vicky Ngo = 90% Mahsa Mohaghegh = 5% Roopak Sinha = 5%

Vicky Ngo

Farzana Zahid

Dr. Mahsa Mohaghegh

Dr. Roopak Sinha

Acknowledgements

A thesis is a formal, written piece of work that embodies results of original research that contribute to the existing body of knowledge. I believe that because there are little rooms to express thanks (and humour) in a scientific piece of work, the acknowledgement section was born as a result. How great is it that this section does not have a word limit! I'm truly glad to have embarked on this research journey. Regardless of the hard works and the outcomes, the last year has not been wasted.

Tremendous thanks to my mom, my younger sister Alisa, and My Phan for their unconditional love and support throughout my life. Without them, I would never have had an opportunity to start this journey, since the day I first come to New Zealand as a Year 7 student until the day I become a postgraduate student. Looking back, it's unbelievable how difficult our entire 5-years journey has been. For that, I would like to express my deepest gratitude.

Dr. Roopak Sinha, thank you for being my supervisor. Your support throughout my journey has been precious to me, ever since my first email to you as a Bachelor's graduate, until today. Thank you for your encouragement in times when I was stuck, and for your reminders to sometimes just slow down and go through things one at a time. Just as you said, research is not about finding the absolute correct answer, but more about the journey itself.

I express my gratitude to Dr. Mahsa Mohaghegh and Dr. Kirill Levchenko for your valuable advice and guidance throughout my research journey as co-supervisors. Even though you are both busy with various commitments, you still made time in your schedule to give me feedbacks on dozen pages long of manuscript.

Thank you to all of my cybersecurity mentors, from whom guidance that my initial research ideas were conceptualised. At that time, I have not the slightest idea about research, and what I knew of cybersecurity was the bare minimum.

I appreciate the companionship that I have had at the Software Engineering Research Center (SERC). I really enjoyed our discussions, from which I gained many great perspectives. And finally, I would like to give thanks to countless other people who have supported me not only in academic matters but also throughout my life.

Vicky Ngo, 3rd January 2023

Chapter 1

Introduction

1.1 Definition of CPS, ICS, and IoT

Cyber-physical systems (CPS) can be broadly defined as an integration of computation, networking and physical process (Lee & Seshia, 2016). CPS have multiple industrial applications in smart grids, industrial control systems (ICS), and manufacturing processes (Khaitan & McCalley, 2015). As such, we could consider ICS to be a type of CPS. Note that because ICS are responsible for controlling different physical components in an industrial setting, ICS is a common target for cyberattacks in the CPS domain due to the effect that a failed ICS could cause on the victim system. Therefore, it is important to also consider cyberattacks in ICS when discussing CPS security.

Internet-of-Things (IoT) involves connecting "things" (objects, devices, machines, etc.) to the internet and to each other (Kiran, 2019). Although IoT shares the same basic architecture with CPS, CPS presents a higher degree of integration between physical and network components. In other words, components in CPS tend to be more tightly connected compared to components in IoT. Because IoT is more concerned with communication and data exchange between different devices primarily through the internet, IoT is more vulnerable to attacks targeting such connections.

1.2 (D)DoS Attacks in CPS and IoT

In CPS and IoT environments, denial-of-service (DoS) and distributed-denial-of-service (DDoS) attacks are known to target its victim systems' resources to bring them down, effectively interrupting the systems operations (Yaacoub et al., 2020; Sharma & Arora, 2021). In the past, cyberattacks on ICS have been mainly motivated by politics, and military causes, but rarely for financial gains. The same also holds for both DoS and DDoS attacks in ICS. However, they have also been historically known to not only act as stand-alone attacks but also as part of multistaged attacks on occasion. During the 2016 Ukraine power grid attack by the CrashOverride (Industroyer) malware, a DoS attack attempted to disable the protective relays in the final stage after interrupting the electricity flow in the victim environment (Slowik, 2019). For context, the protective relays play a crucial role in the power grid self-protection actions, which upon being disabled can potentially cause severe physical damage to the system (Slowik, 2019).

Another well-known malware for DDoS attacks in ICS is Black Energy, used in a multistaged attack in 2015 on Ukraine's power grid. During the attack on the power grid, a DoS attack intercepted the telephone, preventing Ukraine civilians from reporting the outage. According to the Ukraine Ministry of Energy, the outage lasted from 1 to 3.5 hours and created an electricity shortfall of 73 MWh.Overall, past case studies have shown that denial-of-service attacks have had significant impacts on the availability of the systems as well as causing physical damage from a technological viewpoint. As such, the financial loss during the incidents also increases in proportion. The CrashOverride and Black Energy malware described above are only two of the five ICS-tailored malware in the past decade. While the other three malware is not directly relevant in the context of DoS and DDoS attacks on ICS, they are also worth mentioning due to their impacts. These are the Triton/Trisis malware, Havex malware, and the infamous Stuxnet malware. The Maroochy water breach is also a notable cyberattack

on ICS. According to (NCCIC, 2016), DoS was the highest-ranked vulnerability in ICS in 2016.

1.3 Domain Adaptation for Cybersecurity in CPS and IoT

Traditional machine learning models assume that test data and training data have the same data distribution. This assumption is shown in current practice through the fact that most detection models are tested on unseen data from the same dataset chosen for training. Although some models were tested on different datasets, those datasets might ultimately have the same data distribution as the same training dataset, such as in the LUCID (Doriguzzi-Corin, Millar, Scott-Hayward, Martinez-Del-Rincon & Siracusa, 2020) model where the datasets came from the same institution. However, this assumption can be easily violated in real-world applications, as real-world data might have different feature spaces based on where they are generated from. In such cases, the machine learning detection model might not function as desired. This assumption ignores the practicality required for deploying such a model into real-world systems.

To tackle this problem, researchers propose domain adaptation as a solution. The training and test sets are called the source and target domains, respectively. In the context of cybersecurity, domain adaptation aims to apply a trained detection model to a relevant detection task in different domains by minimizing the difference between domain distribution (Farahani, Voghoei, Rasheed & Arabnia, 2020). Considering that training data is also scarce in cybersecurity for CPS and IoT (Gangopadhyay, Odebode & Yesha, 2020), domain adaptation is a promising solution to address this issue.

1.4 Motivation and Significance

Firstly, we seek to assess the (D)DoS attack coverage of existing detection models and datasets in the CPS and IoT domain. After having identified attack types that have not been addressed, we propose the usage domain adaptation to create a cross-domain detection model as a solution to address this issue.

Thus, it is necessary that we investigate existing (D)DoS attack detection models' capabilities for domain adaptation to detect (D)DoS across different domains. Because it is still unclear as to what criteria or requirements would allow a model to perform domain adaptation, we proposed a hypothesis and verify its validity through a controlled experiment. We also identify existing challenges that domain adaptation has on a detection model in terms of each model's learning abilities in different domains, detection performance, and computational overhead, which heavily influence a model's applicability for real deployment.

1.5 Research Questions

RQ1-RQ3 focuses on identifying existing attack types that have not been addressed in both the ICS and CPS domains, given that ICS is also a type of CPS. Because we intend to apply our findings from RQ1-RQ3 to the wider context of CPS in relations to other domains, RQ4-RQ6 would focus more on domain adaptation between (D)DoS detection models in CPS and IoT domains.

RQ1: What are the current state-of-the-art machine learning models for (D)DoS attacks detection in ICS and CPS?

RQ2: How well do the models found in RQ1 cover existing (D)DoS attack techniques in ICS and CPS?

RQ3: What are the current computational challenges in deploying machine learning detection models in ICS, and what contributions have been made to mitigate this?

RQ4: To what extent are existing deep learning models capable of domain adaptation for detecting (D)DoS attacks in CPS/IoT?

RQ5: What are the challenges when adapting machine learning models to CPS/IoT?

RQ6: How are computational overheads affected when adapting machine learning models for detection in CPS/IoT?

RQ1, RQ2, and RQ3 are answered in Chapter 3, while RQ4, RQ5 RQ6 are answered in Chapter 6.

1.6 Contribution

The primary contribution of this research is as follows.

1.6.1 To answer RQ1-RQ3

A systematic mapping study (SMS) between existing (D)DoS attack methods versus current attack detection models and datasets were conducted in 3. We also provided a summary of current state-of-the-art deep learning (D)DoS attack detection models in ICS-relevant domains in terms of optimization parameters and their capabilities to address existing (D)DoS detection techniques. Through the mapping study, we were able to identify current research gaps that need to be addressed, particularly the lack of datasets and models to address numerous existing (D)DoS attacks in the ICS environment. These gaps also apply to other relevant domains of ICS, such as

CPS and IoT. Additionally, we also extend the current (D)DoS attack taxonomy in smart manufacturing systems proposed by (Zahid et al., 2022) to further suit the ICS environment.

1.6.2 To answer RQ4-RQ6

We performed a controlled experiment in domain adaptation on MAD-GAN and LUCID detection models. MAD-GAN is a (D)DoS detection model for CPS, while LUCID is a (D)DoS detection model for IoT. The purpose of the experiment was to investigate the minimum requirements for domain adaptation of detection models. As part of the experiment, we proposed the following hypothesis: **A deep learning (D)DoS detection model is capable of domain adaptation given that it satisfies the following two conditions.**

- **C1 - Transferability of Learned Knowledge;** The model is able to extract domain-relevant features for training and performing detection.
- **C2 - Technical Flexibility:** The detection model source code must allow for alteration in such a way that the model can be easily adapted to datasets from different domains without major changes in its architecture. Thus, hard-coded values that are strictly specific to certain datasets are discouraged.

Our experiment results have identified C1 and C2 as the minimum requirements for domain adaptation of detection models in CPS/IoT. Furthermore, we demonstrated the extent to which domain adaptation affects a model's computational overhead and detection accuracy in a resource-constrained environment, having identified a model's architecture and learning capabilities to be the main influencing factors. Finally, we demonstrated the effect of domain adaptation on a model's learning capabilities when adapted to a CPS/IoT environment and identified a model's feature extraction algorithm

as the key determining factor. Overall, these findings and contributions emphasise the importance of the model's architectural design in determining whether a detection model can be adapted across different domains.

1.7 Thesis Structure

The thesis is structured in manuscript format. Chapter 3 present the first manuscript, which is a systematic mapping study (SMS) of existing datasets and machine learning (D)DoS detection models for ICS, a subset of CPS. Chapter 4 follows by describing our chosen research approach. Chapter 6 reported our experimentation on domain adaptation of (D)DoS detection models in CPS and IoT as manuscript 2. Chapter 2 and Chapter 5 provides an introduction to Chapter 3 and Chapter 6, respectively. Conclusions and future works are presented in Chapter 7.

Chapter 2

Prelude to Manuscript 1

The aim of this chapter is to answer research questions RQ1-RQ3 by identifying existing research gaps through a systematic mapping study. Distributed-denial-of-service (DDoS) attacks pose significant threats to the availability of industrial control systems. In the domain of industrial control systems, denial-of-service attack methods are ever-changing, and zero-day attacks are becoming more prevalent. To combat the threats, researchers have been developing machine learning and deep learning algorithms to help detect attacks in this domain effectively. The usage of such algorithms has been shown to have higher accuracy, precision, and overall detection performance than traditional rule-based methods. However, a comprehensive mapping between the current state-of-the-art and existing (distributed) denial-of-service attack techniques in the Industrial Control System domain is still missing. In this study, we covered 34 common datasets addressing these attacks and 26 state-of-the-art machine learning models for (distributed) denial-of-service attack detection in this domain. We map the datasets and detection models to existing attack techniques, while also identifying research gaps. Based on the results of this mapping study, we discover that the present attack methods are not well addressed by existing datasets and machine learning detection models. The applicability of these models for deployment in real industrial control systems is also

not well-validated because there are so few concrete case studies in the literature.

Ngo, V., Zahid, F., Mohaghegh, M. & Sinha, R. (2022). *A Systematic Mapping of Datasets and Machine Learning Models for (D)DoS Detection in Industrial Control Systems* [Manuscript submitted for publication]. School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology.

Chapter 3

A Systematic Mapping of Datasets and Machine Learning Models for (D)DoS Detection in Industrial Control Systems (Manuscript 1)

3.1 Introduction

Industrial control system (ICS) is a broad class of automation systems used to provide control and monitoring functionality in manufacturing and industrial facilities, which are often considered to be a type of CPS (Knapp & Langill, 2015). Examples of ICS are distributed control systems (DCS), supervisory control and data acquisition (SCADA) systems, and safety instrumented systems (SIS). Because many ICS processes are continuous in nature, any interruption is unacceptable, and thus availability is one of the top priorities (Chhillar, 2020; Stouffer, Pillitteri, Lightman, Abrams & Hahn, 2015). Traditional security measures for ICS include air gaps, proprietary ICS protocols, and security through obscurity (Chhillar, 2020). But as ICS is increasingly integrated

with the internet, these security measures are no longer suitable to protect ICS from cyberattacks, particularly those that impact the system availability, such as denial-of-service (DoS) and distributed-denial-of-service attacks (DDoS). Currently, ICS faces security threats that come with a lack of encryption & authentication, usage of legacy software, and flaws in security policies (Check Point Software Technologies Ltd, 2020). According to recent DDoS analyses, 78,558 DDoS attacks were registered globally in Q2 of 22, with the aviation and aerospace sectors experiencing a 493 percent QoQ (quarter-on-quarter) increase (Yoachimik, 2022; Alexander Gutnikov, Oleg Kupreev & Yaroslav Shmelev, 2022).

As such, researchers have started to consider machine learning (ML) and deep learning (DL) algorithms to assist in cyberattack detection in CPS (Mujeeb Ahmed, Umer, Binte Liyakkathali, Jilani & Zhou, 2021). Through this mapping study, we aim to develop a comprehensive mapping between existing datasets and (D)DoS attack types in ICS to assist in the development of future ML/DL detection models. Additionally, we also performed a mapping between existing detection models for ICS and attack types to provide an overview of current research into ML/DL for ICS and to identify any research gaps. Below, we outline our research questions as follows, which we answer concisely through Tables 3.6, 3.7, 3.8, 3.9.

- **RQ1**. What datasets are available for training and testing ML/DL models in ICS for (D)DoS detection?
- **RQ2**. How well do the datasets in RQ1 cover existing (D)DoS attack types in ICS?
- **RQ3**. How well do currently existing state-of-the-art machine learning and deep learning models cover different types of (D)DoS attacks?
- **RQ4**. What are the potential research opportunities moving forward to improve

current detection models?

In this work, we used the systematic mapping study protocols (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007) to review and analyse existing (D)DoS detection methods using ML and DL approaches and relevant datasets for training such models. We made the following main contributions:

- Provide a mapping between existing industrial datasets and DDoS attack types. This serves as a useful resource for researchers working on ML/DL development for (D)DoS detection in ICS and related domains.
- Based on the datasets used for training and testing, provide a mapping of DDoS detection models for ICS with the attack types that they covered (Section 3.5).
- Extend the taxonomy of (D)DoS attacks in smart manufacturing systems proposed by (Zahid et al., 2022) such that it is more applicable within the ICS context.

This mapping study is organized as follows. Section 3.2 describes the existing surveys on (D)DoS detection methods for ICS-relevant domains, and Section 3.3 lays out the systematic mapping study (SMS) protocols. It is followed by Section 3.4 which presents the mapping between datasets and (D)DoS attack techniques. A similar mapping was performed with state-of-the-art ML/DL detection models in Section 3.5. Finally, Section 3.6 concludes the findings and future directions of this article, as well as answering the research questions.

3.2 Background & Related Works

3.2.1 (D)DoS Attacks and its Impacts on ICS

Denial-of-service (DoS) and distributed-denial-of-service (DDoS) attacks are most notorious for leaving a strong impact on the ICS's availability, especially those that

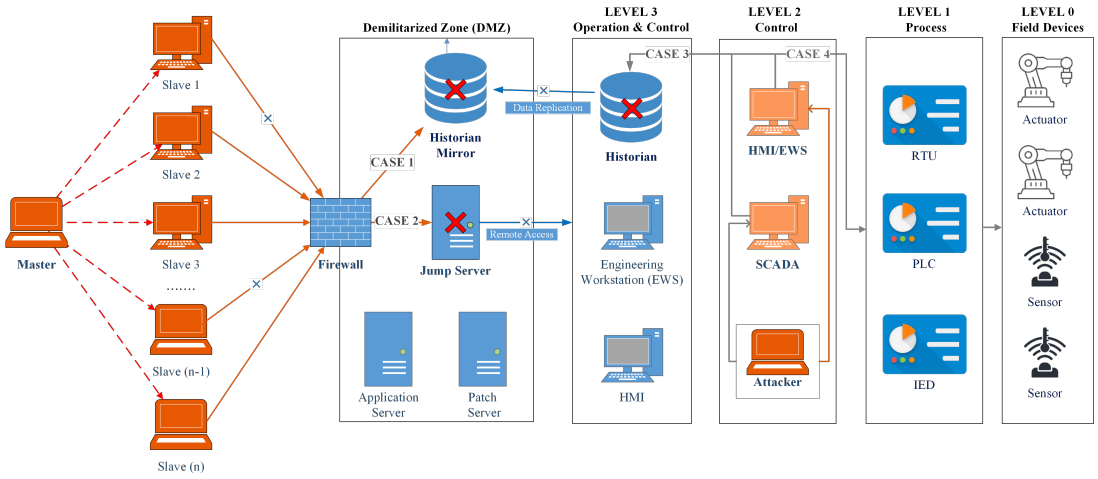

Figure 3.1: Structure of (D)DoS attack cases in ICS

are targeted toward resource-constrained computers (Zahid, Kuo & Sinha, 2021). By definition, DDoS attacks are DoS attacks conducted from multiple systems in synchronisation, all targeting a single victim. Typically, such attacks cause the network to go down or render the systems unable to respond to sensors and actuators, causing heavy damage. (Singh, Yadav & Chuarasia, 2020). Below, we discuss the four possible DoS and DDoS attack cases in ICS with reference to the Purdue Enterprise Reference Architecture (PERA) (Williams, 1993), which is an international standard for designing ICS network architecture that supports operational technology security. Figure 3.1 illustrates these attack cases. There are certainly similarities between DDoS attacks in ICS and in other domains.

Case 1 — DDoS attack on historian mirror

The historian mirror in the demilitarised zone (DMZ) stores copies of data sent from the actual historian in LV3. A DDoS attack on the historian mirror will interrupt the much-needed supply of real-time data. The attack can come from a nearby source outside the network. However, note that the DMZ is an important security feature in the overall PERA structure, where the firewall(s) are located. Thus, it is not an easy task to successfully carry out a DDoS attack in the DMZ (Zeinalpour, 2021).

Case 2 — DDoS attack on jump server

The jump server in the DMZ layer is responsible for securing remote access to devices in the lower levels of the PERA model. While the control and physical processes might not be affected, a DDoS attack on the jump server will lead to denial of access and control at the lower levels. Similar to case 1, the security features in the DMZ will certainly make it difficult for adversaries to attempt attacks.

Case 3 – (D)DoS attack on historian from HMI/EWS

Due to the air-gapped nature of devices from LV3 and below, these attacks tend to occur from within the network. Cyberattackers gain access to the controlling devices (e.g., HMIs, engineering stations) and use them to conduct DoS attacks on the historian through lateral movement. This affects the data mirroring process between the historian and the historian mirror.Further experiments on this attack case were conducted by (Horak et al., 2021; Huraj, Horak, Strelec & Tanuska, 2021) on production line of IIoT systems to observe the effect it has on the historian.

Case 4 — (D)DoS attack on field devices from HMI/EWS

Similar to case 3, cyberattackers utilise lateral movement to perform a DoS attack from the HMI or engineering stations on the field devices, affecting the physical production process itself. Worse, the field devices can be permanently damaged due to the loss of control vulnerability introduced by this attack scenario. Air-gapped security also made it harder to conduct attacks on field devices outside the ICS network. According to (Chan, Chow & Chan, 2019), it is also possible for a reverse process to happen, where the physical devices send attack traffic that interrupts the HMI processes.

Table 3.1: Comparison of existing (D)DoS attacks taxonomies

Ref	Year	Domains	Techniques	Vectors	Effect
(Mirkovic & Reiher, 2020)	2004	Network	No	Yes	Yes
(Zhu et al., 2011)	2011	SCADA	Yes	Yes	Yes
(Wu & Moon, 2017)	2017	CPS	No	Yes	Yes
(Zahid et al., 2022)	2022	CPS	Yes	Yes	Yes

3.2.2 Existing (D)DoS Attack Taxonomies

A comprehensive DDoS attack taxonomy in the relevant field of the ICS is required for mapping, along with detailed information on attack techniques, vectors, and effects. We have found four such taxonomies on major databases, including Scopus, SpringerLink, ACM, and IEEE.

In (Mirkovic & Reiher, 2004), a taxonomy was proposed for categorising DDoS attacks into classes based on their mechanisms, specifically the means to perform the attack, its characteristics, and effect on the victim. However, this study did not describe actual attacking techniques. (Zhu, Joseph & Sastry, 2011) categorised cyberattacks into SCADA based on their target (software, hardware, or communication stacks). The study focuses more on the last category and had limited elaboration on software and hardware attacks. Despite the low number of techniques described overall, each attack technique was described in detail. (Wu & Moon, 2017) proposed a taxonomy consisting of four dimensions: 1) attack vectors, 2) attack impacts, 3) attack targets, and 4) attack consequences. Only the first dimension refers to some actual attack techniques. (Zahid et al., 2022) presented a cross-domain taxonomy specifically for (D)DoS attacks on smart manufacturing systems. The attacks are divided among two classes, *Network* and *Endpoint*, before being categorised into nine difference categories, as seen in figure 3.2. The taxonomy included over 50 attack techniques with their respective attack vectors.

(Mirkovic & Reiher, 2004) and (Zhu et al., 2011) proposed taxonomies that were published in 2004 and 2011, respectively, implying that they may be out of date.

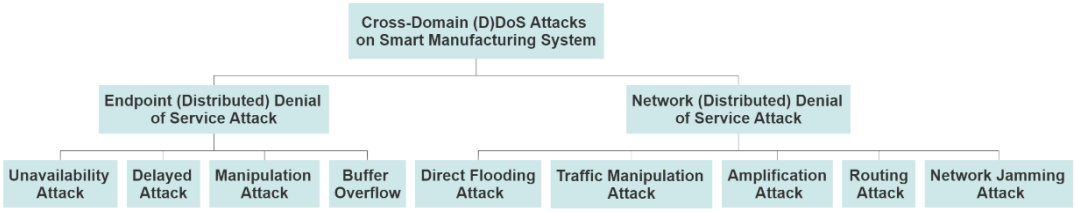

Figure 3.2: Cross-domain taxonomy for (D)DoS attacks on smart manufacturing systems adapted from (Zahid et al., 2022)

Meanwhile, the taxonomy published by (Wu & Moon, 2017) in 2017 does not provide sufficient information on the attack techniques for our mapping study. As the taxonomy proposed by (Zahid et al., 2022) in 2022 is the most up-to-date and comprehensive, we establish this taxonomy as our main reference point for our mapping studies between existing datasets and (D)DoS detection models.

3.2.3 Past surveys of state-of-the-art (D)DoS detection techniques in ICS relevant domains

Seven secondary studies were identified via the SMS process described in Sect. 3.3, which we listed and compared in Table 3.2. The majority of the primary studies chosen for initial screening in this study were published between 2020 and 2022 (Salim, Rathore & Park, 2020; D. Zhang, Wang, Feng, Shi & Vasilakos, 2021; R M Seyam, Bou Nassif, Nasir, Al Blooshi & Abu Talib, 2021; Alimi, Ouahada, Abu-Mahfouz, Rimer & Alimi, 2021; Arora, Kaur & Teixeira, 2022; Tama, Lee & Lee, 2022; J. Zhang et al., 2022).

Only the DoS attack taxonomy introduced by (Salim et al., 2020) includes detailed attack techniques, however the taxonomy only includes common DDoS attacks on the cloud, targeted toward IoT domain rather than ICS-relevant domains.

Following the DDoS attack taxonomy introduced by (Salim et al., 2020), the authors have also performed a mapping between the detection models and the taxonomy. However, only a handful of ML/DL models were included. Another three

Table 3.2: Comparison of previous surveys on intrusion detection techniques for ICS relevant context using ML/DL

Ref	Domains	Datasets Map	Model Map	DoS Taxon.	Datasets Used
(Salim et al., 2020)	IoT	No	Yes	Yes	No
(Alimi et al., 2021)	ICS	No	No	No	Yes
(Seyam et al., 2021)	ICS	No	No	No	Yes
(D. Zhang et al., 2021)	ICPS	No	No	No	No
(Arora et al., 2022)	ICS	No	No	No	Yes
(Tama et al., 2022)	ICS	No	No	No	Yes
(J. Zhang et al., 2022)	CPPS	No	No	No	Yes
Our work	*ICS*	*Yes*	*Yes*	*Yes*	*Yes*

DoS Taxon. = DoS Taxonomy
Perf. = Detection Performance - Whether the study have included information on its chosen models' detection performance
Method Desc. = Method Description - Whether the study have included the techniques used by its chosen model for attack detection.

studies, (R M Seyam et al., 2021; Arora et al., 2022; J. Zhang et al., 2022), included the types of attacks detected by each of the models in their reviews. None of these three studies referenced any (D)DoS attack taxonomy, therefore it was difficult to find a common reference point to perform a mapping. Furthermore, the mapping carried out by (R M Seyam et al., 2021; Arora et al., 2022; J. Zhang et al., 2022) was not aimed at (D)DoS attacks but at a broader range of attack categories, including DoS, reconnaissance, probing, and false data injection. Overall, we conclude that there is a lack of mapping between existing detection models and (D)DoS attacks in ICS and relevant domains.

In addition to the lack of mapping between detection models and (D)DoS attacks, we observed a similar lack of mapping between datasets and (D)DoS attack types. (Alimi et al., 2021) is the only work that provided information on the attack types covered by commonly used datasets for (D)DoS detection. Nevertheless, the dataset mapping provided by (Alimi et al., 2021) only included the general attack categories in each dataset instead of detailed techniques and was not focused on (D)DoS attacks. In conclusion, the information by the aforementioned taxonomy was not sufficient to map

the datasets with (D)DoS attack methods.

Out of the seven primary studies, (Alimi et al., 2021) and (J. Zhang et al., 2022) are the most comprehensive as they both include both techniques and datasets used by each model, as well as their performance and evaluation metrics. In fact, the latter work has the most thorough description of each model detection techniques. Meanwhile, (D. Zhang et al., 2021) focused on attack detection and estimation from a control perspective, where system modelling and complexity analysis became more important. As such, (D. Zhang et al., 2021) did not include most ML/DL relevant metrics, as shown in Table 3.1.

It should be noted that (Tama et al., 2022) provided a very thorough review of the models chosen for the studies, as well as making a remark for each one. However, these remarks mostly mentioned possible improvements instead of including what each model was best at. As such, it was difficult to assess which detection model was more superior.

(R M Seyam et al., 2021) was the only one to provide a distribution of commonly used datasets in the reviewed models, as well as commonly used performance metrics. However, the work also includes numerous detection models for network security, which might not represent detection models in ICS relevant domains.

3.3 Systematic Mapping Study (SMS) Protocols

A systematic mapping study (SMS) is a broad review of primary studies for the purpose of analysing and classifying existing literature in a particular domain (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007). A mapping study helps identify areas where further primary studies or systematic literature reviews are needed (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007). The outcome of the SMS process for this work will be a taxonomy

of existing high-performing machine learning and deep learning methods for (D)DoS attack detection in ICS, as well as a taxonomy of relevant datasets for training such models. A mapping will be performed between the detection models and datasets with existing (D)DoS attack techniques.

3.3.1 Search Strategies

The databases chosen for the literature review are IEEE, SpringerLink, Science Direct, ACM Digital Library, and Scopus. This is because SpringerLink, Science Direct, and Scopus are major publishers of peer-reviewed journals in the science field. IEEEXplore and ACM are also high ranking journals based on the Scimago Journal Rank. Additionally, Google Scholar was also used as a secondary source to ensure that other relevant studies outside the aforementioned publishing venues and databases were not excluded.

During this search process, we used search terms such as "denial of service", "industrial control system", "cyber-physical systems", "machine learning", "deep learning", "detection", "model", "review", along with their synonyms and alternate spellings. This search string below is one such example, which we appropriately adjusted for each search engine used.

(Detection OR classification) AND (injection OR intrusion OR DDoS OR DoS OR denial-of-service OR denial of service) AND (deep learning OR machine learning) AND (methods OR algorithms OR model OR technique) OR (cyber physical systems OR CPS OR industrial control systems or ICS or SCADA) AND (survey OR review OR mapping)

3.3.2 Inclusion and Exclusion Criteria

We used two sets of inclusion and exclusion criteria in this mapping study: one for selection of datasets, and one for selection of studies and past surveys.

Datasets Selection

To ensure that the dataset included is relevant in the context of ICS, we first make sure that it meets all the inclusion criteria and none of the exclusion criteria.

Inclusion Criteria:

- **IC1:** The dataset must include at least one type of (D)DoS attacks presented in section 3.2. For example, the ADFA dataset was not included in this table because it only contains traffic relevant to the context of web browsers.
- **IC2:** The dataset must have been cited by papers on ML/DL development for (D)DoS attack detection in the context of ICS, CPS, IIoT, IoT or WSN.
- **IC3:** The dataset must be obtainable in some ways, either through being publicly available or through request with the authors. The authors' intention for sharing their datasets can be found in various sources, such as through the sites or their research papers on the datasets.

Exclusion criteria:

- **EC1:** The dataset was not intended specifically for training of (D)DoS attack detection model.

The types of attacks in each dataset were identified through multiple methods. We primarily obtain this information through the dataset creators and their respective sites or papers. In cases where this method is not possible, we examine studies that have used those datasets before instead, which often include the attack types from their chosen datasets. Likewise, we also obtain other information about the datasets, such as their sizes, through the same methods.

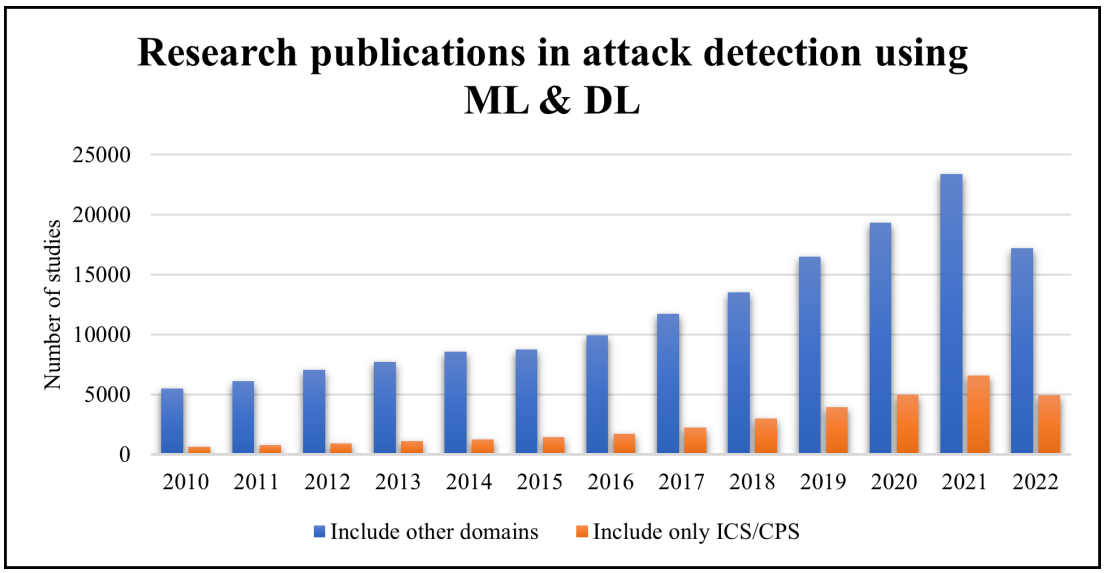

Figure 3.3: Research publications in attack detection using ML & DL

For ML & DL models Selection

Due to the large number of studies that could be obtained after the initial search, it is impractical to examine the full text of each candidate. Thus, a set of inclusion and exclusion criteria should be established to filter out studies that do not help answer the research questions. A study is excluded upon satisfying any of the exclusion criteria. Note that this set of criteria only applies to the initial stage of screening primary and secondary studies. Through further inspection of each study, we aim to narrow down to approximately 20–30 of the best-performing detection models to include in this study using the "Final Selection" criteria, as described further below.

According to Google Scholar, the number of research papers on attack detection using machine learning in ICS & CPS and other domains has been rising significantly since 2016. This data is shown in Figure 3.3 which documents the number of studies per year from Q1 2010 to Q3 2022. The numbers include both primary and secondary studies.

Screening Inclusion Criteria:

- **S-IC1**: For studies that propose new techniques for attack detections using ML and/or DL, the published time range should be from 2016 to 2022. For secondary studies, they should be published from 2019 to 2022, which allows them to cover studies from previous years.
- **S-IC2**: If the publication is outside the specified timeframe, they can only be included upon consideration of the following.
  - The extent to which the work is still being updated after its publication
  - Applicability for use in development of other models
  - Number of citations for their positive contribution
- **S-IC3**: The chosen study must be specifically targeted toward detection within ICS-relevant domains.

Screening Exclusion Criteria:

- **S-EC1**: Studies that are not in written in English, or studies that do not have full text available.
- **S-EC2**: Studies published before the specified time range.
- **S-EC3**: Studies on detection models in non-ICS-relevant domains, attacks prevention and mitigation methods are out of scope for this study.

Below, we present the final selection criteria which decides whether a model should be selected into the final set of studies. A model does not necessarily meet all inclusion criteria, however it must not meet any final exclusion criteria listed.

Final Inclusion Criteria:

- **F-IC1:** High detection performance achieved through an optimized design or novel optimization technique (s).

- **F-IC2:** Consideration for real-life application, such as lightweightness, real-time detection, integrability and scalability into existing system, or other novel features.
- **F-IC3:** Validated results and claim of significance through justified comparison and validation method(s).

Final Exclusion Criteria:

- **F-EC1:** No claim of significance. The authors did not prove how their studies differentiate from other similar studies in terms of performance, optimization approaches, attack types addressed, detection model architecture.
- **F-EC2:** A lack of validation of results. The authors did not sufficiently justify how the machine learning technique used and its respective performance validation technique were effective.
- **F-EC3:** Lack of proof for applicability. There are minimal indications of how the model can be applied for similar attacks outside the tested dataset(s). This proof can be evaluation conducted on a testbed or through testing the model's performance on different datasets.

The initial search returned 1,583 articles, of which 13 were secondary studies. 7 of these secondary studies were chosen as per the screening criteria. Using both the chosen secondary studies and the initial search, 176 primary studies were identified for initial screenings through title, abstract, and keywords. Using S-IC1 and S-IC2 resulted in 116 studies to be examined. After careful examination of the introduction, conclusion, and proposed model's architecture and performance, 26 primary studies were included in the final set. Fig. 3.4 describes the overall literature search and selection process.

We also took notes of the datasets used by the relevant studies during the search and review processes. Additionally, we also conducted additional searches of existing

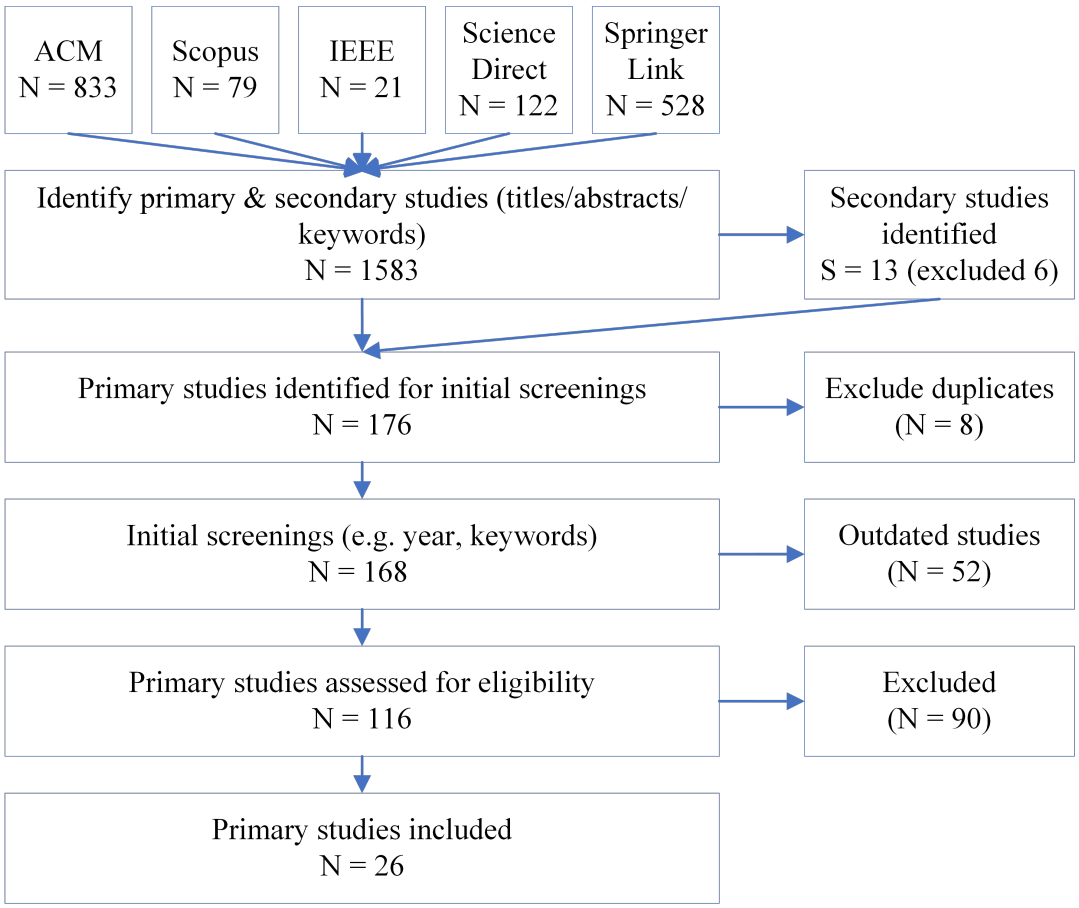

Figure 3.4: Literature search and selection

datasets for attack techniques listed in the (D)DoS attack taxonomy proposed by (Zahid et al., 2022). Using IC1-3, 34 datasets for training (D)DoS attack detection models in ICS were included in the final set.

3.3.3 Data Extraction

The data extraction processes follow the searching process, aiming to record findings or information obtained from the studies in a structured form such that they contribute to answering the research questions (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007). For each study, the form should include the name of the reviewer, date of data extraction, title, authors, journal, publication details,

and additional notes (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007). Additionally, it is also suggested that a second extraction be performed on random primary studies and cross-checked against the first extraction (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007).

In our studies, the information extracted is as follows for each study: authors, source (DOI), model types, datasets used, attack types addressed, strengths and weaknesses, optimization techniques, and performance metrics.

3.3.4 Quality Assessment

Quality assessment is a process where primary studies relevant to answering the research questions are evaluated. It allows for more detailed inclusion and exclusion criteria to be included, as well as the weighting of individual studies when synthesising data (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007). By critically reviewing the research, bias can be minimised while maximising internal and external validity. The following questions review whether the research questions were addressed adequately (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007). The quality assessment questions were scored with either "Yes", "No", or "Partly" (*Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007).

3.4 Datasets Mapping for ML/DL

In our mapping study for existing datasets with (D)DoS attack types, we used (Zahid et al., 2022)'s (D)DoS taxonomy as our main reference point. In addition to the existing attacks listed in the nine existing categories (unavailability, delay, manipulation, buffer

Table 3.3: Quality assessments for the research questions

Quality Assessment Questions	Evaluation
Q1: Are the review's inclusion and exclusion criteria described and appropriate?	(Y) The inclusion and exclusion criteria are clearly defined before starting the literature review, which aims to include as many relevant resources as possible on existing techniques for both ML and DL.
Q2: Is the literature search likely to have covered all relevant studies?	(Y) In addition to searching in four major databases (Scopus, Springer, Science Direct, Google Scholar), three specific journals were also referenced (ACM, IEEE, Nature Portfolio). Search strategies, including usages of boolean AND and OR, are also employed for each database.
Q3. Did the reviewers assess the quality/validity of the included studies?	(P) Due to time efficiency and numerous studies found, only the introduction, abstract, and conclusion of each study were addressed. For studies that reported valuable findings, their methodologies, results, and literature review section were also considered.
Q4. Were the basic data/studies adequately described?	(P) Only a summary of certain aspects of each of the studies were included.

overflow, direct flooding, traffic manipulation, amplification, routing, and network jamming), we expanded the aforementioned taxonomy by adding the following additional attack techniques to some categories, as shown in Fig. 3.4 for *Endpoint* attacks and Fig. 3.5 for *Network* attacks. The new addition allows us to ensure full coverage for all attack techniques in each dataset and to categorise them more effectively.

Through our literature review process, we selected 33 frequently used datasets for (D)DoS attack detections in ICS and CPS, using the criteria described in Sect. 3.3. Using the same set of criteria, we identified different types of attacks in each dataset and mapped them into the different categories proposed in (Zahid et al., 2022)'s taxonomy. This allowed us to accurately determine the current coverage of existing datasets.

Aside from identifying the types and categories of attacks covered for each dataset,

Table 3.4: Additions to endpoint (D)DoS attacks in taxonomy proposed by Zahid et al. (2022)

Sub-Class	Names	Type	Mode	Vector(s)
Unavailability Attack	Puppet attack	DoS	Indirect	Select node as puppet node to flood AMI with RREQ packets
Delayed Attack	Replay attack	DoS	Direct	Maliciously replays/repeats valid transmission
Manipulation Attack	Complex Malicious Response Injection	DoS	Indirect	Device configuration, device model, open ports, service information, threshold values information, run-time configuration, temporal features correlated with device physics and operations
	Malicious state command injection			
	Malicious function code command injection			
	Malicious parameter command injection			
	Complex Malicious Response Injection			
Buffer Overflow	Memory corruption attack	DoS	Direct	Exploits memory corruption vulnerabilities to execute malicious code, crashing program and corrupting data
	Heap overflow	DoS	Direct	Overwrite data in heap, execute arbitrary code to crash program
	Stack exhaustion	DoS	Direct	Overwrite data in stack, execute arbitrary code to crash program
	CPU exhaustion	DoS	Direct	Exploits CPU denial-of-service vulnerability; use of malicious packets

Table 3.5: Additions to network (D)DoS attacks in taxonomy proposed by Zahid et al. (2022)

Sub-Class	Names	Type	Mode	Vector(s)
Direct Flooding Attack	ICCP[a] Attack	DoS	Direct	Use of large number of TCP connection requests to overload the ICCP server
Traffic Manipulation Attack	PCCC[b] Attack	DoS	Direct	Sending malicious Execute PCCC service to trigger Major Error (0x08)
	Ethernet/IP Attack	DoS	Direct	DoS packet; EtherNet/IP Register Session Request; Connection Manager Forward Open Request
	CIP[c] Attack	DoS	Direct	Malicious CIP packe, abuse of CIP commands for improper access control leading to IP changes, triggered reset, or fault state
	DNP3 Attack	DoS	Direct	Fabricated/malicious messages, DFC flag, malformed packets
	Fieldbus Attack	DoS	Direct	Lack of authentication, placing transmitter to “active” state
	GOOSE[d] Attack	DoS	Direct	Lack of encryption & authentication, manipulated GOOSE frames, use of spoofed packets
	CAN[e] bus Attack	DoS	Direct	Injecting high-priority messages in a short time
	S7Comm Injection	DoS	Direct	Lack authentication, use of spoofed packets, hijacking
	TCP Reset (RST)	DoS	Direct	Send forged TCP RST packets to force stop TCP connections
Amplification Attack	LDAP[f]	DDoS	Indirect	Use LDAP server to amplify DDoS attacks
	TCP	DDoS	Indirect	Abuse TCP retransmission to perform DoS;
	Modbus master traffic jamming	DoS	Direct	Continuously transmit random data to random destination address

[a] Inter-Control Communication Centre Protocol, [b] Programmable Controller Communication Commands, [c] Common Industrial Protocol, [d] Generic Object Oriented Substation Event, [e] Controller Area Network, [f] Lightweight Directory Access Control

we also attempted to determine their sizes and whether the attacks recorded were real or synthetic through relevant studies. While a large dataset will provide more data from the same source for detection models to be trained or tested upon, there are also performance differences between using real versus synthetic traffic. Further findings on this topic are also covered below.

3.4.1 Attack Coverage Provided by Existing Datasets

In the development of machine learning and deep learning models, datasets play a crucial role in deciding the models' detection performance. Therefore, obtaining relevant and up-to-date datasets is a common concern. Most models that we observed used multiple datasets to test, train, and evaluate their proposed models. These datasets include both public datasets and traffic data collected from the authors' own test beds. The traffic may or may not be collected from real devices, emulated or simulated traffic, or test beds developed by the authors.

The feature matrix presented in Table 3.6) summarised each dataset coverage of attacks from (Zahid et al., 2022)'s taxonomy, Fig. 3.4 and Fig. 3.5. Frequently used datasets in existing studies were also bolded. The feature matrix allows us to observe the diversity of coverage for the attack categories elaborated in (Zahid et al., 2022)'s taxonomy. Thus, a 'Y' does not necessarily indicate that all attack types in that category are covered.

Further observation of the attack types covered has allowed various discoveries with regard to the attack coverage provided by existing datasets. Note that a dataset size classification is dependent on the number of packets and file size when compared to other similar datasets, not necessarily the diversity of attack types. However, these two variables are often correlated. Below, we elaborate on each category, from the most well-covered to the least sufficiently covered.

Manipulation Attacks Each of the attack types in the manipulation attack categories is supported by at least two datasets, with most of them being large and well-cited. Only data aggregation attack is not supported. Overall, manipulation is the most balanced and well-supported category.

Table 3.6: Mapping of datasets in ICS with (D)DoS attack categories

		Endpoint				Network				
Year	Dataset	Unavailability	Delayed	Manipulation	Overflow	Flooding	Traffic	Amplification	Routing	Jamming
1998	**NSLKDD**	Y	N	N	Y	Y	N	Y	N	N
1999	**KDD99**	Y	N	N	Y	Y	N	Y	N	N
2000	DARPA99	Y	N	N	Y	Y	N	Y	N	N
2006	Kyoto 2006	N	Y	N	Y	N	N	N	N	N
2007	CAIDA DDoS	N	N	N	Y	Y	Y	N	N	N
2012	**ISCX2012**	N	N	N	Y	N	Y	N	N	N
2012	TUIDS2012	Y	N	N	Y	Y	Y	N	N	N
2014	**Water Tank**	N	N	Y	N	N	Y	N	N	Y
2014	Booter DNS2014	Y	N	N	N	N	Y	Y	N	N
2015	SANTA	N	N	N	Y	Y	N	Y	N	N
2015	**Gas Pipeline**	N	N	Y	N	N	Y	N	N	Y
2015	**Power System**	N	Y	Y	N	N	N	N	N	N
2015	**UNSW-NB15**	Y	N	N	Y	Y	Y	Y	N	N
2015	**SWaT**	N	N	Y	N	Y	Y	Y	N	N
2016	**BATADAL**	N	Y	N	N	N	N	N	N	N
2016	BDD	N	N	N	N	N	N	N	Y	N
2017	**CICIDS2017**	N	N	N	Y	Y	N	N	N	N
2017	**WADI**	N	N	Y	N	N	Y	N	N	N
2018	**CSE-CIC2018**	Y	N	N	Y	N	Y	N	N	N
2018	WSN-DS	N	N	N	Y	N	N	N	Y	Y
2019	**CICIDS2019**	Y	N	N	Y	N	N	Y	N	N
2019	Bot-IoT	Y	N	N	Y	Y	N	N	N	N
2019	DNP3-Snort	N	N	N	N	N	Y	N	N	N
2019	Electra	N	Y	Y	N	N	Y	N	N	N
2019	IEC61850Security	N	N	N	N	N	Y	N	N	N
2019	Edge-IIoT	Y	N	N	Y	Y	Y	Y	N	N
2019	Modbus TCP SCADA	Y	N	N	Y	N	N	N	N	N
2019	Modbus dataset	N	N	N	Y	Y	Y	N	N	N
2020	BOUN DDoS 2020	Y	N	N	Y	N	N	N	N	N
2020	RPL-Based Iot	N	N	N	N	Y	N	N	N	N
2021	WUSTL-IIOT-2021	N	N	Y	N	N	N	N	N	N
2021	EPIC	N	N	Y	Y	N	N	N	N	N
2022	MSCAD	N	N	N	Y	Y	N	N	N	N

Y (Green) = Yes, at least one attack type is included.
N (Red) = No, the dataset doesn't include any attack type in that category.

Buffer Overflow Attacks Buffer overflow attacks are also supported by multiple large, well-cited, up-to-date datasets, such as the CICDDoS2019, CSE-CIC2018, CICIDS2017, and ISCX2012. Only the denial of sleep and CPU exhaustion attack types were not presence in any datasets.

Unavailability Attacks Similar to buffer overflow category, unavailability attacks are covered by multiple high-quality datasets. Permanent DoS and denial of message attacks were not covered. Interestingly, the included puppet attack type was a novel attack reported by only one research paper, which also proposed a method to detect it. Therefore, the lack of supporting dataset for the puppet attack should not have an

adverse effect on the overall coverage of this category.

Amplification Attacks In this category, most attacks are addressed by large, well-cited datasets. QUIC and DTLS amplification attacks that have not been included in any datasets thus far, and only DARPA dataset cover for TCP-Resets attack. Meanwhile, the CIC-DDoS2019 dataset that covers five out of the eight attack types in this category.

Network Jamming Attacks All five attack types in the network jamming attack category were addressed, except for the sporadic jamming attack. However, it is alarming to see that the constant, random, and reactive jamming attacks were all covered by only the WSN-DS dataset, a rather small dataset, while the gas pipeline dataset only covers Modbus master traffic jamming. It is clear that the development of attack detection models against jamming attacks will require more datasets to truly validate the models' performances.

Direct Flooding Attacks The ICMP flood, also known as "smurf attack", is covered by numerous attack sets as it is a very common DDoS in various domains. Meanwhile, the hello flood attack is only addressed by the RPL-based IoT dataset, with only the TUIDS dataset covering for the QUIC flood attack. Unfortunately, none of the other attack types were included in any datasets.

Traffic Manipulation Attacks The seemingly large amount of coverage for the traffic manipulation attack category was actually caused by the inclusion of HTTP flood attacks in multiple datasets. This is because HTTP flood is a very common attack that occurs in numerous domains, including web applications, not just ICS and CPS. Only half of the industrial communication protocol attacks in this category are covered. Even so, only one or two small datasets would address each type of attack in this category, displaying a clear lack of datasets for communication protocol-specific attacks.

Delayed & Routing Attacks The last two attack categories are also the least well covered, which are delayed attacks and routing attacks. In the delayed attack group, only replay attack is covered. The other four attack type in the same category, as shown in (Zahid et al., 2022)'s taxonomy, do not have any datasets available. For routing attack, only black hole attack is considered by the BDD and WSN-DS datasets. Note that the BDD dataset was developed only for black hole attack alone, and that both datasets are of small sizes.

In summary, we can conclude the following regarding the availability of datasets for each attack category:

- The most well-covered attack categories are **Manipulation, buffer overflow, and unavailability**. This is shown through the attack types' presence in large, well-cited datasets.
- The **amplification** attack category is quite sufficiently covered, except for the QUIC and DTLS attack types that have not been included in any datasets. The jamming attack category requires more datasets to support its attacks aside from the WSN-DS dataset.
- The coverage for **direct flooding and traffic manipulation** attack categories is extremely biassed toward certain attacks, which are ICMP flood and HTTP flood for each category, respectively. The majority of the other attack types in these two categories are not sufficiently addressed or do not make it to any dataset at all.
- **Delayed and routing** attacks are virtually not supported at all, with many attacks showing strong favour for the replay attack in the delayed category and no supporting datasets for other attack types except for the BDD dataset for black hole attacks.

In our findings, NSL-KDD, KDD CUP 99(KDD99), and DARPA are the three

datasets that are most referred to in various studies, due to their usefulness in the past. As the aforementioned datasets become more outdated, newer datasets from the Canadian Institute for Cybersecurity (CICIDS2017, CSE-CIC2018, CICDDoS2019), iTrust lab (SWaT, WADI, EPIC), and University of New Brunswick (UNSW-NB15, BoT-IoT) are becoming more well-known among researchers. Specifically, the SWaT dataset is frequently cited for its application in the CPS, ICS context, with datasets from the other two institutions being applied in more diverse domains such as web application, edge computing, ICS, CPS, etc.

Through observing Table 3.6, it is shown that the most commonly used datasets are (e.g. CICIDS2019) for DDoS attacks. For general intrusion detection models, they mostly use NSL-KDD 99 and KDD99 datasets, in addition to data collected or generated from other sources. Both the KDD99 and DARPA 1998/99 are considered old datasets are considered old because they have not been updated with the most recent attack patterns (Mittal, Kumar & Behal, 2022; "A Review of Intrusion Detection Systems: Datasets and Machine Learning Methods", 2021). For detection models in ICS, the SWaT, WADI, and gas pipeline dataset are common.

One special exception is the MSCAD dataset, which was only made available in April 2022 (citation). The dataset is quite new, and thus it is not surprising to see that it is perhaps the less known dataset in the table. However, it is still included due to its potential applicability to the ICS context.

3.4.2 Datasets Characteristics & Availability

When selecting a dataset for model training, two important factors should be considered: how large the dataset is and whether the generated traffic was simulated or from a real test bed. Due to the difficulty of gaining access to large-scale real-world systems, dataset generation will inevitably involve using test beds most of the time. However,

this does not necessarily imply that all datasets were simulated. In our mapping study, a dataset is considered real if it came from test beds that resembled the actual systems close enough to generate realistic traffic. This means that the realistic test beds will not include a network simulator. For example, the ISCX2012 (Shiravi, Shiravi, Tavallaee & Ghorbani, 2012),CICIDS2017 (Sharafaldin., Habibi Lashkari. & Ghorbani., 2018), CSE-CIC2018 (Communications Security Establishment (CSE) & Canadian Institute for Cybersecurity (CIC), 2018), and CICIDS2019 (Sharafaldin, Lashkari, Hakak & Ghorbani, 2019a) datasets were all generated using profiles based on real human behaviours on the network.

Characteristics such as packet counts, total recorded duration, and file size can be used to categorise dataset sizes. While file size might seem to be a suitable metric for measuring dataset sizes, it is in fact influenced by other factors such as file formats, compression tools used, and types of attacks performed. Thus, we faced the issue of size inconsistencies in the datasets. For example, although the ISCX2012 dataset contains 1,796,753 samples (Khan, Karim & Kim, 2019) and the WUSTL-IIoT-2021 dataset contains 1,194,464 samples (Zolanvari, Teixeira, Gupta, Khan & Jain, 2021), the ISCX2012 data is 69.6 GB (Khan et al., 2019) in size while the WUSTL-IIoT-2021 dataset size is only 2.7 GB (Zolanvari et al., 2021). Because of how frequently these inconsistencies appeared, file size should not be used as an indicator of dataset size. On the other hand, most machine learning models detect anomalies using packet analysis, implying that packet count is a better metric for measuring dataset sizes. Since packet counts and total recorded duration tend to be directly correlated, we use the following categorization criteria in classifying dataset sizes, in order of precedence.

- Observation counts: Below 500,000 (Small), 500,000 - 1,500,000 (Medium), over 1,500,000 (Large).
- Total recorded duration: 0h - 12h (Small), 12h - 24h (Medium), over 24h (Large).

Finally, it is worth noting that all the datasets listed in table 3.6 are all obtainable in one way or another, which in fact was one of the selection criteria. The detailed information can be seen in table 3.7. In the table, the links to the official page or dataset source for the datasets are provided. Typically, these links will also include information on the relevant published papers about these datasets. Note that:

- A **'Yes'** implies that the dataset is directly accessible through the provided source.
- A **'Request'** implies that the datasets will be accessible through making a request with the author(s). Further information on this can be found on the sources provided in Table 3.7 for each dataset. In most cases, either a request form or direct contact, possibly through email, with the author(s) is required.
- A **'Restricted'** implies that the dataset link provided, is restricted, possibly due to firewall or geo-blocking. A possible workaround is to make a request to the author(s).
- A **'No'** implies that no information was found regarding the dataset source, however it is likely that the author(s) is willing to share. This is indicated by the dataset respective research paper explicitly 'proposing a new dataset'. It is possible to contact the author(s) for further access.

3.5 ML/DL Detection models for (D)DoS attacks

Following the review protocols, we have included a final set of 26 studies. We mapped these models to (D)DoS attack categories proposed by (Zahid et al., 2022) and our addition in section 3.4, which is shown in table 3.9.

Table 3.7: Availability of datasets for (D)DoS attack detection in ICS

Year	Dataset	Real	Size	Public	Source
1998	**NSLKDD**	No	Small	Yes	https://www.unb.ca/cic/datasets/nsl.html
1999	**KDD99**	No	Large	Yes	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
2000	DARPA99	No	Large	Yes	https://archive.ll.mit.edu/ideval/docs/attackDB.html
2006	Kyoto 2006	Yes	Large	Yes	https://www.takakura.com/Kyoto_data/
2007	CAIDA DDoS	No	Small	Request	https://www.caida.org/catalog/datasets/ddos-20070804_dataset/
2012	**ISCX2012**	Yes	Large	Yes	https://www.unb.ca/cic/datasets/ids.html
2012	TUIDS2012	Yes	Medium	Yes	http://agnigarh.tezu.ernet.in/~dkb/resources.html
2014	Booter DNS2014	Yes	UNK	Yes	https://www.simpleweb.org/wiki/index.php/Traces
2014	**Water Tank**	Yes	Small	Request	https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets
2015	**Gas Pipeline**	Yes	Small	Yes	https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets
2015	**Power System**	No	Small	Yes	https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets
2015	SANTA	Yes	UNK	No	https://doi.org/10.1109/BIBE.2014.72
2015	**SWaT**	Yes	Medium	Request	https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/
2015	**UNSW-NB15**	No	Large	Yes	https://research.unsw.edu.au/projects/unsw-nb15-dataset
2016	**BATADAL**	No	Large	Yes	https://www.batadal.net/data.html
2016	BDD	No	Small	No	https://www.jucs.org/jucs_22_4/feature_selection_for_black/jucs_22_04_0521_0536_yassein.pdf
2017	**CICIDS2017**	Yes	Large	Yes	https://www.unb.ca/cic/datasets/ids-2017.html
2017	**WADI**	Yes	Medium	Request	https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/
2018	**CSE-CIC2018**	Yes	Large	Yes	https://www.unb.ca/cic/datasets/ids-2018.html
2018	WSN-DS	No	Small	Yes	https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds
2019	Bot-IoT	No	Large	Yes	https://research.unsw.edu.au/projects/bot-iot-dataset
2019	**CICIDS2019**	Yes	Medium	Yes	https://www.unb.ca/cic/datasets/ddos-2019.html
2019	DNP3-Snort	Yes	Small	Yes	https://github.com/igbe/DNP3-Dataset-Plus-SnortRules
2019	Edge-IIoT	Yes	Large	Yes	https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot
2019	Electra	Yes	Large	Restricted	http://perception.inf.um.es/ICS-datasets/
2019	IEC61850Security	No	Medium	Yes	https://github.com/smartgridadsc/IEC61850SecurityDataset
2019	Modbus dataset	No	Large	Yes	https://github.com/antoine-lemay/Modbus_dataset
2019	Modbus TCP SCADA	Yes	Small	Yes	https://github.com/tjcruz-dei/ICS_PCAPS/releases/tag/MODBUSTCP%231
2020	RPL-Based Iot	No	Small	Request	https://wise.cs.hacettepe.edu.tr/projects/rplsec/
2021	EPIC	Yes	Small	Request	https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/
2021	WUSTL-IIOT-2021	Yes	Medium	Yes	https://www.cse.wustl.edu/~jain/iiot2/index.html
2022	MSCAD	No	Small	Yes	https://www.kaggle.com/datasets/drjamailalsawwa/mscad

UNK (Yellow) = Unknown

3.5.1 State-of-the-arts ML/DL models for (D)DoS detection

In this mapping studies, we categorize the chosen models based on the optimization made on them into 4 main categories, with further details as shown in Fig. 3.5.

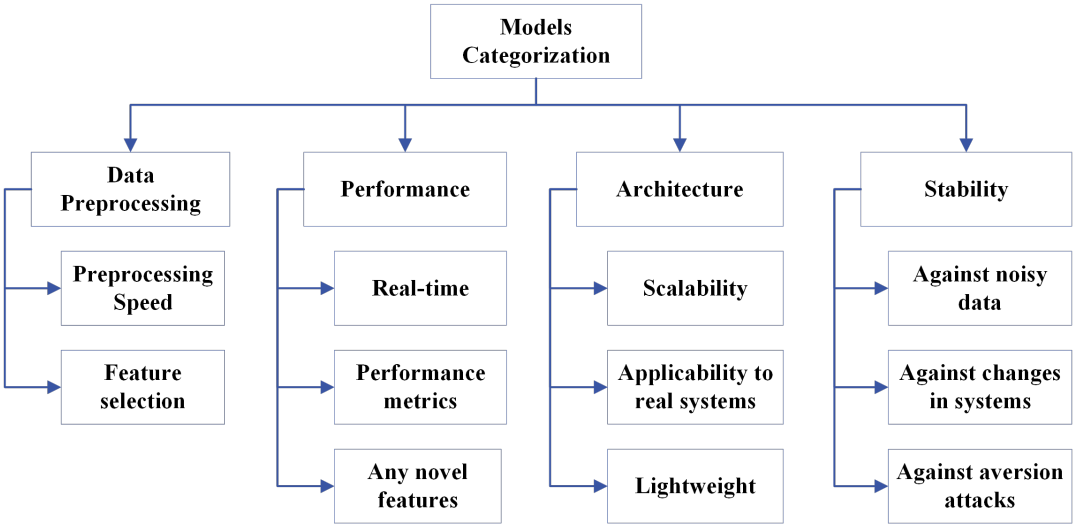

Figure 3.5: Categorization of ML/DL (D)DoS detection models

Table 3.8: State-of-the-art ML/DL models for (D)DoS detection in ICS and CPS

Author	Year	Model	Optimisation	Domain	Datasets	Performance Metrics	Remarks
(Doshi, Apthorpe & Feamster, 2018)	2018	NN	Architecture	IoT	Testbed	ACC, F1, PRE, TPR	High scalability
(Lin, Adepu, Verwer & Mathur, 2018)	2018	TABOR	Architecture	CPS	SWaT	CP, FPR, PS, TPR	High applicability, can works as simulation controller
(Wang et al., 2019)	2019	AE	Architecture	CPS	Testbed	ACC, F1, PRE, TPR	Scalable and compatible with different systems
(Kim, Jo & Shon, 2020)	2020	APAD	Architecture	IIoT	SWaT	ACC, FPR, TPR	Lightweight
(Hassan, Gumaei, Huda & Almogren, 2020)	2020	Ensemble	Architecture	IIoT	Power System	ACC, FPR	Scalable, can be combined with existing detection engines
(Demertzis, Iliadis & Bougoudis, 2020)	2020	Gryphon	Architecture	ICS	Gas Pipeline, Power System, Water Tank	F1, PRE, TA, TNR, TPR	Low cost and running time, lightweight
(Doriguzzi-Corin, Millar, Scott-Hayward, Martínez-del Rincón & Siracusa, 2020)	2020	LUCID	Architecture	IoT	CICIDS2017, CSECIC2018, ISCX2012	ACC, F1, FPR, PRE, TPR	Lightweight, fast preprocessing speed
(Gao et al., 2021) 2021 FNN-LS	2021	FNN-LSTM	Architecture	ICS	KDD99	F1, PRE, TPR	Minimize individual weaknesses found in FNN & LSTM
(Fährmann, Damer, Kirchbuchner & Kuijper, 2022)	2021	LW-LSTM-VAE	Architecture	ICS	SWaT, WADI	F1, PRE, TPR	Lightweight, short training time
(He, Mendis & Wei, 2017)	2017	CDBN	Performance	CPS	Testbed	ACC, PRE, TPR	Real-time detection
(Feng, Li & Chana, 2017)	2017	LSTM	Performance	ICS	Gas Pipeline	ACC, F1, PRE, TPR	High detection performance

Author	Year	Model	Optimisation	Domain	Datasets	Performance Metrics	Remarks
(Yang, Cheng & Chuah, 2019)	2019	CNN	Performance	ICS	Testbed	ACC, PRE, TPR	Has re-training scheme incorporated
(H. A. Khan, Sehatbakhsh, Nguyen, Prvulovic & Zajic, 2019)	2019	DNN	Performance	CPS	Testbed	ACC, FPR, TPR	Able to monitor target device remotely
(I. A. Khan, Pi, Khan, Hussain & Nawaz, 2019)	2019	HML-IDS	Performance	ICS	Gas Pipeline	ACC, F1, Kappa coefficient, PRE, TPR	High detection performance
(Derhab et al., 2019)	2019	RSL-KNN	Performance	IIoT	Power System	ACC, FPR	Using blockchain gives high detection performance
(Y. Li & Wang, 2020)	2020	Caps-GN	Performance	ICS	Testbed	ACC, PRE, REC	Real-time detection. Can identify attacking device location. Stable detection rate
(Ferrag & Maglaras, 2020)	2020	DeepCoin	Performance	CPS	Bot-IoT, CICIDS2017, Power System	ACC, FPR, TPR	Using blockchain gives high detection performance
(Das, Adepu & Zhou, 2020) 2020	2020	LAD-based	Performance	CPS	SWaT	F1, PRE, TPR	Real-time detection, lightweight
(S & D, 2017)	2017	HNA-AA	Preprocessing	ICS	ADFA-LD, MIRD, MORD, Power System, SIRD, SORD, Testbed	ACC, Dice similarity, Jaccard index, FPR, PRE, TNR, TPR	High detection performance
(McDermott, Majdani & Petrovski, 2018)	2018	BLSTM-RNN	Preprocessing	IoT	Testbed	ACC, loss	Novel technique for feature selection
(Schneider & Böttinger, 2018)	2018	NDAE-DNN	Preprocessing	IIoT	Modbus dataset, SWaT	F1, PRE, TPR	Optimise packet acquisition speed, overcame loss of interpretable features

Author	Year	Model	Optimisation	Domain	Datasets	Performance Metrics	Remarks
(Yan, Mestha & Abbaszadeh, 2019)	2019	ELM	Preprocessing	CPS	Testbed	FPR, TPR	Novel technique for salient features identification
(Hachimi, Kaddoum, Gagnon & Illy, 2020)	2020	C-RAN	Preprocessing	WSN	WSN-DS	ACC, FPR, TPR	Optimized traffic preprocessing
(AL-Hawawreh, Moustafa & Sitnikova, 2018)	2018	DAE-DFFNN	Stability	IIoT	KDD99, UNSW-NB15	ACC, FPR, TPR	Tolerate noisy and outlier data
(Abdelaty, Doriguzzi-Corin & Siracusa, 2020)	2020	AADS	Stability	ICS	SWaT	F1, PRE, TPR	Robust to noisy data, adapt to changes in ICS normal behaviour
(Kravchik & Shabtai, 2022)	2021	CNN-AE	Stability	ICS	BATADAL,SWaT,WADI	F1, PRE, TPR	Resilient to adversarial evasion attacks.

As seen in table 3.8, most studies optimise their models' architecture or performance. A smaller number of studies optimise their data preprocessing technique and therefore improve their models' detection performance. Only 3 studies have models' stability as their main focus. We've also discovered that the majority of detection models employ an anomaly detection approach, in which the models are trained on normal traffic data to detect attacks as any deviation from normal behaviour. This allows for the detection of zero-day attacks as opposed to signature-based detection. We also noticed that deep learning techniques take up the majority of studies reported, particularly RNN, AE, or other neural networks, with some studies combining two algorithms into a single detection system for better performance. Accuracy (ACC), precision (PRE), true positive rate (TPR), and F1 score are most often used performance metrics.

Most studies chose the gas pipeline, power system, and water tank datasets for model training. The datasets all came from the Center for Cybersecurity Research and Education (CCRE) at the University of Alabama (Morris, n.d.). SWaT, BATADAL, WADI are also well-known datasets for attack detection in ICS, with SWaT being more frequently used. Other studies typically use traffic data from their own testbeds.

Studies that optimise model architectures tend to be lightweight or scalable, or otherwise have shorter training and execution times. Real-time detection and high performance metrics are often the focus of performance optimization, with some models possessing additional novel features aside from attack detection. 5 models optimised its data preprocessing performance through feature selection, which often results in better performance and more lightweight model. Only 3 models optimise performance stability, with 2 of them reported to be robust against noisy data and the last one being resistant against evasion attacks.

Among the reported studies, we've identified many models with novel features. Two models used blockchain and its immutability to optimise detection performance (Ferrag & Maglaras, 2020; Derhab et al., 2019). The model proposed by (Y. Li & Wang,

2020) was able to identify location, direction, and connection of the attacking device, presumably within the same network. Another model increased its packet acquisition speed through parallel pipe lining (Schneider & Böttinger, 2018).(Abdelaty, Doriguzzi-Corin & Siracusa, 2020; H. Yang, Cheng & Chuah, 2019) works were designed so that they can be easily updated with new traffic data. Particularly, the model proposed by (H. Yang et al., 2019) has a re-training scheme to be triggered at a certain number of new attacks, meanwhile being able to adapt to any SCADA environment.

Nevertheless, we also find gaps in the current research area regarding (D)DoS detection in ICS using ML and DL.

- ***Lack of testing for attacks outside of training datasets*** Most studies train and test their models using the same dataset(s). This means that only a limited number of attacks in those datasets were used to evaluate the models' performance. Although the anomaly detection approach theoretically allows models to detect unseen attacks, there are minimal empirical results to support this hypothesis. Furthermore, testing detection models against different datasets will help evaluate their applicability for deployment in real systems, where the traffic data differs from that in the training datasets.

- ***Lack of framework for modelling testbeds for detection performance evaluation***. The majority of the models were evaluated against other cutting-edge technologies, alternative datasets, and/or simulated traffic. Some studies deployed their suggested models onto their own testbeds for analysis and testing. In order to properly assess the efficacy of the models' detection, it is crucial that the testbed accurately reflects the real systems and software currently employed in the industry. It is a challenging task given that there is no established framework for evaluating such testbeds.

- ***Lack of model deployment in real systems***. In all of our reported studies, there

have not been any case studies of deploying models in real ICS systems. While there are many external factors and complications that may be involved in doing so, it is still necessary to conduct such tests when possible to evaluate the detection models against actual, real-time traffic.

As for the models that we have excluded during the literature review protocols, there were three main reasons:

- ***Low dataset diversity***. Some models were trained and tested with the same datasets. Thus, these models might not perform well with data from other sources.
- ***Using old datasets***. The most notable examples are the usage of datasets like NSLKDD, KDD99, and DARPA in some models despite their high detection performance.
- ***Lack of validation***. Some studies tested and trained with the same datasets, while others only compared their models with different types of algorithms but not state-of-the-art solutions. These studies do not mention whether the authors developed the comparison algorithms or if they sourced them elsewhere, meaning there is a possibility of unintentional bias.

The lack of machine learning models' deployment into real ICS brings into question the validity and applicability of these models themselves. Therefore, we have also conducted external searches for in-depth case studies of such models' usage in real CPS for cyberattack detection. To our best knowledge, only two companies, SparkCognition (*Artificial Intelligence Solutions*, 2022) and Uptake (*Industrial Intelligence*, 2022) are known to offer AI solutions for cyberattack detection in industrial settings. As businesses, detailed case studies of the relevant AI solutions are often kept confidential. In spite of the few published use cases present in the literature, this finding

Table 3.9: Mapping of (D)DoS detection models in ICS with (D)DoS attack categories

			Endpoint				Network				
Author	**Year**	**Detection Model**	**Unavailability**	**Delayed**	**Manipulation**	**Overflow**	**Flooding**	**Traffic**	**Amplification**	**Routing**	**Jamming**
[30]	2018	NN	Y	N	Y	Y	N	N	N	N	N
[31]	2018	TABOR	N	N	Y	N	Y	Y	Y	N	N
[32]	2019	AE	N	N	Y	N	N	N	N	N	N
[33]	2020	APAD	N	N	Y	N	Y	Y	Y	N	N
[34]	2020	Ensemble RT	N	Y	Y	N	N	N	N	N	N
[35]	2020	Gryphon	N	Y	Y	N	N	Y	N	N	Y
[36]	2020	LUCID	Y	N	N	Y	Y	Y	N	N	N
[37]	2021	FNN-LSTM	Y	N	N	Y	Y	N	N	N	N
[38]	2021	LW-LSTM-VAE	N	Y	N	Y	Y	Y	N	N	N
[39]	2017	CDBN	N	N	Y	N	N	N	N	N	N
[40]	2017	LSTM	N	N	Y	N	N	Y	N	N	Y
[41]	2019	CNN	Y	N	N	Y	N	Y	N	N	N
[42]	2019	DNN	N	N	N	Y	Y	N	N	N	N
[43]	2019	HML-IDS	N	N	Y	N	N	Y	N	N	Y
[44]	2019	RSL-KNN, BICS	N	Y	Y	N	N	N	N	N	N
[45]	2020	Caps-GN	N	N	Y	N	N	N	N	N	N
[46]	2020	DeepCoin (RNN)	Y	Y	Y	Y	Y	N	N	N	N
[47]	2020	LAD-based	N	N	Y	N	Y	Y	Y	N	N
[48]	2017	HNA-AA	N	Y	Y	N	N	N	N	N	N
[49]	2018	BLSTM-RNN	Y	N	N	Y	N	N	Y	N	N
[50]	2018	NDAE+DNN	N	N	Y	Y	Y	Y	Y	N	N
[51]	2019	ELM	N	N	Y	N	N	N	N	N	N
[52]	2020	C-RAN	N	N	N	Y	N	N	N	Y	Y
[53]	2018	DAE-DFFNN	Y	N	N	Y	Y	Y	Y	N	N
[54]	2020	AADS	N	N	Y	N	Y	Y	Y	N	N
[55]	2021	CNN & AE-based	N	N	Y	N	Y	N	Y	N	N

Y = Yes, at least one attack type is addressed by the model
N = No, the model doesn't address any attack type in that category

demonstrates that machine learning detection models for industrial environments are in fact commercially available for deployment in limited numbers.

3.5.2 Attack coverage of state-of-the-art models

Through this mapping study, we have found that detection models' attack coverage is nearly entirely dependent on the attack datasets they use, except for using test beds. While utilising anomaly detection techniques would allow detection of unknown attacks, models are not often tested for different types of attacks outside the chosen datasets in their studies. We have made the following observations for each category, from the most well covered to the least well covered.

Manipulation Attacks Despite the low dataset coverage for this category, Table 3.9 seemingly showed a larger coverage. This is in fact due to the large usage of the Water

Tank, Power System, and Gas Pipeline datasets, which covered most of the data and command injection attacks in this category. However, the fact that most studies use the same datasets without variation indicates a lack of datasets for manipulation attacks. To address this, some studies have used test beds and carried out their own attacks instead.

Traffic Manipulation & Direct Flooding Attacks Most models cover these two categories through the SWaT and Gas Pipeline datasets, specifically the Modbus attacks and TCP reset attacks. Other datasets for general networks such as KDD99, CICIDS2017, CSECIC2018, and UNSW-NB15 were used; however, they only include common attacks such as HTTP flooding. The reported models rely nearly entirely on public datasets' data, meaning that there is little coverage for other protocol specific attacks and other flooding attacks in existing detection models.

Buffer Overflow Attacks The reported models covering this attack category use testbeds half the time. It is difficult to know what types of attacks have been conducted on these test beds unless stated in the paper. Other models rely on a diverse range of datasets, including those for ICS (e.g., SWaT, WADI) and networks (e.g., CICIDS2017, CSECIC2018, KDD99). Note that ICS datasets often only cover common attacks such as TCP SYN floods or slow rate attacks. UNSW-NB15 and ISCX2012 datasets combined cover most buffer overflow attacks. However, these two datasets were not often used in the reported models. No model has explicitly addressed the denial of sleep attacks, resulting in approximately half of the category not considered for.

Amplification Attacks Amplification attacks are one of the least well covered category, as seen with the low number of models shown in Table 3.9. Furthermore, nearly all of them trained on the SWaT dataset, which only covers the TCP resets attack. None of the reported models used the CICIDS2019 dataset, which covers amplification

attacks best.

Unavailability Attacks Only 6 out of 26 reported models addressed unavailability attacks, either through test beds or through datasets for the general network such as KDD99, CICIDS2017, CSECIC2018, and UNSW-NB15 datasets. It is interesting to note that NSLKDD, KDD99, and DARPA datasets have already covered most of this attack category. However, the chosen studies most likely did not use them because they are already outdated. This suggests a lack of up-to-date dataset for unavailability attacks.

Delayed, Routing, and Network Jamming attacks As shown in Table 3.9, these 3 categories are the least well covered. This is because existing datasets have very little coverage of attacks in these categories, as shown in Chap 3.4.

To summarise, manipulation attacks are mostly covered by well-known datasets in ICS. Only a small fraction of attack techniques in the traffic manipulation and direct flooding attack categories were addressed, despite the seemingly large coverage. Amplification and buffer overflow attacks have high-quality datasets ready for use and publicly available; however, the reported models did not use these datasets, which results in much lower coverage. The unavailability attacks category also does not have any up-to-date datasets addressing it. Finally, delayed, routing, and network jamming attacks are the least well covered. Even studies that used test beds barely addressed attacks in these three categories.

3.6 Conclusion & Future Works

There exists a lack of public, up-to-date datasets for most (D)DoS attack techniques addressed in these studies, especially protocol-specific attacks and less common attacks

such as delayed, routing, and network jamming attacks. We find that some studies have been using outdated datasets such as KDD99, NSLKDD to address certain (D)DoS techniques because there are no other datasets addressing them. It is also critical that the data gathered came from real ICS networks or equivalents rather than simulation. For less well covered attacks, current datasets should also be sufficiently large.

On the other hand, we've identified common issues with existing detection models as follows, not only from the studies included in the work but also from other studies that we excluded when conducting the literature review.

- A majority of the models that we have come across through the SMS protocol only test their proposed detection models using existing datasets, while the rest use testbeds and/or compare detection performance with existing detection algorithms. To our best knowledge, there have not been any reported case studies of deploying such detection models in real ICS networks or systems.

- Models often do not test for attacks outside the chosen datasets or those that their models have been trained on. As such, it is crucial that future studies also address this to ensure an accurate evaluation of the proposed detection models in changing environments.

- Only a few models consider lightweight and real-time detection capability as one of their optimising parameters, which have been included in these studies. Given that legacy systems do not handle the addition of new detection algorithms or software very well, the development of lightweight models should be examined. Computational overhead should also be part of the consideration.

- There is a need for research on detection models for other less-covered attack types, especially those not covered in any datasets. These include protocol-specific injection attacks, routeing attacks, and delayed attacks.

Additionally, none of the studies mentioned any framework or standards that used when developing testbeds. This suggests a lack of such frameworks or standards for doing so. Such a framework would be useful to ensure testbeds' qualities and representativeness of real-world systems. Not only this be helpful for research in attack detection models for ICS, but also for any other studies that require similar testbeds in the academic environment where access to real-world, industry-standard systems is not available.

We also find that there are few in-depth case studies on deployment of machine learning models for cyberattack detection in actual ICS. Similar models and AI solutions have been demonstrated to be commercially available for industrial settings, although detailed reports on use cases and detection performance are very limited in the literature.

3.6.1 Threats to Validity

We outline the main limitations of this mapping study.

- ***Language bias***. As only models in the English language are included, it is possible for other well-developed models written in other languages to be excluded.
- ***Time period exclusion***. Only models and reviews from 2016 until the present are considered in this literature review. Furthermore, the majority of the selected models that performed well were published in 2020. Thus, some models that were published earlier might be excluded.
- ***Time constraints***. Due to time constraints, it is more efficient to identify significant research through published secondary studies. Thus, not all studies appearing in other form of publication can be examined. Furthermore, only DL models are considered in this review.
- ***Reviewer bias***. There exists possible reviewer bias because there were only

resources for one primary researcher to apply inclusion and exclusion criteria. Terms such as"high detection performance" and "justified results" can have meanings subjective to different researchers. Therefore, the inclusion criteria are applied liberally to make the review as objective as possible.

- ***Assumption of Completeness***. For most ML and DL models presented, specific types of attacks were not mentioned. Instead, the authors typically refer to the datasets or test beds used. Therefore, a major assumption in the review is that a model was trained on all attacks in any given dataset, unless specified otherwise.

- ***Exclusion of detection models from other domains***. In this study, we have not taken into consideration other (D)DoS attack detection models from other domains that could be applied to the ICS context.

- ***Exclusion of studies on optimisation techniques***. Since this mapping study on existing state-of-the-art detection models, we consider papers that propose optimisation techniques for these models to be out of scope.

3.6.2 Future Works

Future works to address research gaps identified in this mapping study includes but not limited to:

- Development of ML/DL detection models for DoS/DDoS attacks that have not been addressed, such as attacks targeting security flaws in ICS communication protocols in the traffic manipulation attacks category, as well as delayed and routing attacks.

- Optimisation or improvement on existing ML/DL detection models toward deployment in real industrial systems. This includes consideration for features

such as lightweightness, real-time detection, scalability, maintainability, datasets diversity. For performance validation, we encourage that the proposed models be given trial runs on suitable test beds.

- Development of new datasets for attacks that have yet to be addressed, especially for the CPS and ICS domains, given that current datasets lack coverage for delayed, routing, and protocol-specific attacks.

- Development of a framework for ICS test beds to ensure that they are representative of the real world ICS network, and would enable accurate evaluation of attack detection models in this domain.

An extension to this mapping study could be a comprehensive mapping between each model and the attack techniques, instead of just attack categories. Additionally, we could also include detection models from other domains instead of using only models developed specifically for CPS and ICS. Inclusion of further studies on optimisation techniques for ML-based detection systems or algorithms would also be helpful.

Chapter 4

Research Method

Our mapping study results in Chapter 3 have identified an ongoing lack of up-to-date datasets of current (D)DoS attack techniques to train detection models. Approaches to address this issue include developing new datasets, or using transfer learning to fine-tune a pre-trained model for a specific detection domain, especially in domains that do not have abundant data available. Domain adaptation is an example of transfer learning techniques that have been used for training detection models in cybersecurity. Domain adaptation is a tangible solution in our case given that it already has proven applications in other domains such as computer vision and natural language processing. Based on our understanding of domain adaptation in cybersecurity, the following hypothesis is proposed.

A deep learning (D)DoS detection model is capable of domain adaptation given that it satisfies the following two conditions.

- **C1 - Transferability of Learned Knowledge;** The model is able to extract domain-relevant features for training and performing detection.
- **C2 - Technical Flexibility:** The detection model source code must allow for alteration in such a way that the model can be easily adapted to datasets from

different domains without major changes in its architecture. Thus, hard-coded values that are strictly specific to certain datasets are discouraged.

This section discusses the research approach and strategy that we used to test this hypothesis, which we further describe in Sec. 6.1.

4.1 Choosing a Research Strategy

A research strategy is described as methods used to systematically produce a solution to a given research problem (Kothari, 2004). It is important to establish the chosen research plan and methodology clearly before conducting the actual research work. Likewise, selecting an appropriate research strategy is an equally important task. Below, we examined six different research approaches while considering how to design and evaluate our hypothesis (Easterbrook, Singer, Storey & Damian, 2008).

- **Controlled Experiments**: A controlled experiment involves investigating a testable hypothesis where the relationship between the independent and dependent variables is explored (Easterbrook et al., 2008). This is done by making controlled changes to the independent variables and observing the effect these changes have on the dependent variables. It is important that the experiments be conducted in a controlled environment, where other external factors aside from the independent variables must not be allowed to influence the experiment results. This is to ensure the validity of the cause-effect relationship established upon observation of the dependent variables (Easterbrook et al., 2008).

- **Case Studies**: A case study can be understood as an empirical enquiry that seeks to explain why a certain phenomenon occurs within a real-life context and reveal any relevant cause-and-effect relationships (Easterbrook et al., 2008). Case study

research uses purposive sampling to select the most relevant cases to the study proposition, but it is also more open to researcher bias (Easterbrook et al., 2008).

- **Survey Research**: Survey research is a quantitative method to collect information from a representative sample, from which generalisations can be drawn and applied to the wider population (Easterbrook et al., 2008). Survey research often involves the use of questionnaires and different sampling methods to collect data. However, it is important to also control for sampling bias to ensure that the selected sample is representative of the larger population (Easterbrook et al., 2008).

- **Ethnographies**: Ethnography focuses on field observations of a target community in their own environment, and thus has a more sociological focus (Easterbrook et al., 2008). While no pre-existing theories are assumed for ethnography research, it is crucial to avoid preconceptions. In the context of software engineering, ethnography would be helpful for studies focusing on aspects of particular technical communities, such as work practises (Easterbrook et al., 2008).

- **Action Research**: In action research, the researcher studies methods to solve an existing problem, while also documenting and analysing the process of solving that problem (Easterbrook et al., 2008). Therefore, action research tends to imply organisational commitment that comes with implementing the solution proposed and observing its impacts (Easterbrook et al., 2008).

- **Design Science**: Design science approach involves the study and creation of an artefact for a specified problem domain (Hevner, March, Park & Ram, 2004). The artefact must be rigorously evaluated, and proven to solve the specified problem in a more effective manner that is of interest to the general population (Hevner et al., 2004). Additionally, the design science research must be communicated

effectively to both technical and managerial audiences.

Through careful evaluation of the six proposed research approaches, we considered the controlled experiment method to be the most fitting to evaluate our hypothesis.

4.2 Controlled Experiments

A controlled experiment is an investigation of a testable hypothesis where the effect of manipulating independent variables on dependent variables are evaluated (Easterbrook et al., 2008). Controlled experiments allow us to determine whether a cause-effect relationship exists between these variables, and how they are related to each other. As such, the precondition of a controlled experiment is a clear and testable hypothesis, which will guide the subsequent steps in the experiment as well as deciding what variables are to be included. In a controlled experiment, there are three important types of variables:

- **Independent Variables**: Independent variables are those that will be changed across the experiment to observe the effect that these changes have on the dependent variables. In our experiments, the independent variables are the models to be trained, and the datasets that each model was trained and evaluated with.

- **Dependent Variables**: Dependent variables are variables whose value depends upon changes made by independent variables. We seek to find if a cause-effect relationship exists between the dependent and independent variables. These are the performance metrics of the trained models upon evaluation, including computational overhead.

- **Controlled Variables**: Any variables other than the chosen independent variables that can affect the dependent variables are considered controlled variables. It

is crucial not to let the controlled variables affect the experiment results to ensure a fair experiment. In this study, the main controlled variables include the consistency of the datasets and the evaluation environment used. The extent to which we modify the chosen models' source code must also be controlled to reliably adapt the models without affecting their core architecture and intended purposes.

Additionally, the method in which we evaluate the experiment empirical validity is also of importance. (Easterbrook et al., 2008) identify the following four criteria for validity:

- *Construct validity* focuses on whether theoretical constructs and abstract definitions are interpreted and measured correctly. For example, different researchers may have different interpretations of the terms "detection performance" and "computational overhead." As a result, we have explicitly defined the meanings of these two terms in Sect. 6.3.

- *Internal validity* focuses on the study design and whether the results follow from the data. Allowing uncontrolled variables outside independent variables to affect dependent variables is a common threat to internal validity.

- *External validity* focuses on whether the generalisation of results and subsequent claims are justified. Specifically, how do the results of domain adaptation experiments on our chosen models generalise to the hypothesis, and whether such generalisation is justified?

- *Reliability* focuses on whether the experiment is reproducible and whether the same results can be achieved if other researchers replicate the study design. To maximise reproducibility, we have described the experiment's implementation in

detail in Sect 6.4, as well as made the source code we used and other technical details publicly available in (Ngo, Mohaghegh & Sinha, 2022).

Chapter 5

Prelude to Manuscript 2

Domain adaptation and transfer learning have applications in computer vision and natural language processing, where machine learning models are adapted for use in similar tasks across different domains. Domain adaptation allows existing detection models to leverage existing knowledge toward solving a specific but similar problem, and also addresses the lack of labelled training data, which is an ongoing concern for (D)DoS detection models in cyber-physical systems and Internet-of-Things. However, there is currently minimal research into adapting existing detection models in different domains and the challenges involved. In this study, we stated a hypothesis on the minimum conditions required for domain adaptation of a detection model across different detection domains. To verify our hypothesis, we performed domain adaptation on two (D)DoS attack detection models in Internet-of-Things and cyber-physical systems, respectively. The two models were tested on a resource-constrained environment to evaluate their detection performance and computational overhead. The results following domain adaptation of the two models show that our hypothesis holds true. We discover that the detection models architecture determines what types of information that these models can learn from. Depending on whether this knowledge is transferable to another domain, the models' detection accuracy will be affected. Further work is required to

validate how well our hypothesis generalises to cybersecurity in Internet-of-Things and cyber-physical systems domains.

Ngo, V., Mohaghegh, M. & Sinha, R. (2023). *Domain Adaptation of Deep Learning (D)DoS Attack Detection Algorithms for CPS and IoT* [Manuscript submitted for publication]. School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology.

Chapter 6

Domain Adaptation of Deep Learning (D)DoS Attack Detection Models for CPS and IoT (Manuscript 2)

6.1 Introduction

In 2006, Dr. Helen Gill from the United States National Science Foundation coined the term "cyber-physical systems" (CPS) (Lee & Seshia, 2016). Specifically, CPS can be broadly defined as the integration of communication, control, and software components into physical processes. CPS have multiple industrial applications in smart grids, industrial control systems, and manufacturing processes (Khaitan & McCalley, 2015). In such cases, CPS operations tend to be continuous in nature and often have higher availability requirements. Therefore, interruptions to these processes can be detrimental to the system's operations and, in rare cases, can cause serious damage to the physical devices. As such, denial-of-service (DoS) and distributed-denial-of-service (DDoS) attacks pose great cybersecurity concerns for CPS.

IoT is defined as the interconnection of physical objects, or "things," embedded with

sensors, software, and other technologies that enabled them to collect and exchange data over the internet (Kiran, 2019). IoT is often seen in smart homes, healthcare, and other industrial applications, which are commonly referred to as Industrial Internet-of-Things (IIoT). IoT infrastructure is susceptible to (D)DoS attacks that threaten IoT device security (Ibrahim, Abu Al-Haija & Ahmad, 2022).

IoT and CPS are similar in that they are both used to control physical processes through network communication. Common (D)DoS attack techniques in IoT have been shown to overlap with those in CPS (Zahid et al., 2022). Thus, a (D)DoS detection model trained for IoT environment would also be useful for detection in CPS environment, and vice versa, comparing to knowledge transfer from models trained in unrelated fields.

In the literature, various (D)DoS detection algorithms based on ML and DL have been proposed for both IoT and CPS domains. However, a majority of detection algorithms in CPS consider neither the actual applications of these models in resource-constrained systems, nor sufficiently address existing (D)DoS attack techniques targeting CPS (Ngo, Zahid, Mohaghegh & Sinha, 2022). According to (Ngo, Zahid et al., 2022), the lack of up-to-date datasets of current (D)DoS attack techniques to train detection models is also a major factor. Transfer learning, or domain adaptation, of machine learning and deep learning detection models are common methods to address the lack of training data, mainly in natural language processing, computer vision and cybersecurity. Given that CPS and IoT are closely related domains, it is possible to use domain adaptation of (D)DoS attack detection models in IoT for use in the CPS domain and vice versa. Thus, we propose the following research questions:

> **RQ1**: To what extent are existing deep learning models capable of domain adaptation for detecting (D)DoS attacks in CPS/IoT?
>
> **RQ2**: What are the challenges when adapting machine learning models to CPS/IoT?

RQ3: How are computational overheads affected when adapting machine learning models for detection in CPS/IoT?

To answer the research questions, we performed a controlled experiment in domain adaptation of existing deep learning (D)DoS detection models, particularly MAD-GAN and LUCID (D. Li, Chen et al., 2019; Doriguzzi-Corin et al., 2020). The two models were chosen based on four main criteria: 1) context-relevance to deep learning and domain adaptation, 2) domain-relevance to CPS and IoT, 3) credibility and citations, and 4) source code availability. We identified how domain adaptation affects the models' performance as well as the challenges involved.

In our study, we primarily considered domain adaptation of the first type according to (Pan & Yang, 2010), where the feature spaces of the source and target domains are different. This scenario is frequently encountered in real-life detection since incoming traffic might not always have the same feature spaces as the training datasets' feature spaces. While domain adaptation inherently requires that a model be able to transfer and apply its training to a different domain, it is unclear which technical criteria would facilitate domain adaptation in a detection model, as well as why certain models are unable to perform domain adaptation. Based on this understanding, we propose the following hypothesis, whose validity we sought to verify through our experiment result.

A deep learning (D)DoS detection model is capable of domain adaptation given that it satisfies the following two conditions.

- **C1 - Transferability of Learned Knowledge;** The model is able to extract domain-relevant features for training and performing detection.
- **C2 - Technical Flexibility:** The detection model source code must allow for alteration in such a way that the model can be easily adapted to datasets from different domains without major changes in its architecture. Thus, hard-coded values that are strictly specific to certain datasets are discouraged.

There are two possible and mutually inclusive outcomes in our experiment that would prove the hypothesis to be true. The first outcome allows for domain adaptation of the model by satisfying both conditions. The second outcome covers the cases where either or both of the conditions were not satisfied, leading to the chosen model being unable to perform domain adaptation. The third outcome refers to the case where a model is capable of domain adaptation without having to satisfy both conditions; in this case, our hypothesis will be disproven. Through performing the experiments set out in this study, we made the following contributions:

- Identified C1 and C2 as the minimum requirements for domain adaptation of (D)DoS attack detection models for IoT and CPS environments.
- Demonstrated the effect of domain adaptation on a detection model's computational overhead and detection accuracy in resource-constrained environments, whereas the model's architecture and learning capabilities are the main influencing factors, respectively.
- Demonstrated the models' learning capabilities when adapted to CPS/IoT, having identified the models' feature extraction algorithm as the key influencing factor.

This study is organized as follows. Section 6.2 describes existing (D)DoS detection algorithms and domain adaptation for CPS and IoT. Section 6.3 continues with our experiment design. Then, the implementation details are presented in Section 6.4. Finally, Section 6.7 concludes the findings and future works.

6.2 Background

6.2.1 Deep Learning Based (D)DoS Attack Detection Models in CPS/IoT

Deep learning (DL) algorithms are known to achieve better accuracy than machine learning algorithms by utilising neural networks (Tama et al., 2022). In recent years, researchers have started to consider machine learning (ML) and DL algorithms for cyberattack detection in CPS (Mujeeb Ahmed et al., 2021). Deep neural network algorithms, such as RNN, CNN, and autoencoder, have seen frequent usage for such purposes (Tama et al., 2022). Similar results were found in a systematic mapping study of state-of-the-art deep learning models for industrial control systems, where CNN, autoencoders, and LSTM (a type of RNN) are common (Ngo, Zahid et al., 2022). These deep learning models often employ the anomaly detection approach, which is highly effective against false data injection attacks and malware in the CPS domain (Tama et al., 2022). A few algorithms use a signature-based approach, with a small percentage taking advantage of both approaches (Tama et al., 2022).

Deep learning detection models also have various applications in IoT and smart homes. It is worth noting that the (D)DOS attack types present in the IoT domain are vastly different compared to (D)DoS attacks in CPS. Particularly, attacks in IoT include application layer attacks, botnet attacks, and transport layer attacks. These attacks are common not only in IoT but also target web applications and computer networks (Salim et al., 2020). In a survey of recent attack detection models in IoT by (Babu & Veena, 2021), variations of neural networks and RNNs are most common in addressing these attacks. We also discovered that the majority of the models reported in (Babu & Veena, 2021) follow a semi-supervised training approach, using both labelled and unlabelled data to train the detection models. Furthermore, the three algorithms CNN, ANN, and

LSTM have been shown to have returned the best detection performance in terms of accuracy for the IoT domain (Ahmad & Alsmadi, 2021).

6.2.2 Domain Adaptation of Deep Learning (D)DoS Algorithms

Traditional deep learning algorithms assume that the feature space of the training data (source domain) is the same as the feature space of the data on which the model is applied (target domain) (Gangopadhyay et al., 2020). In fact, common practice usually splits a chosen dataset into training, testing, and validation sets to train and evaluate a machine learning model, whereas the feature spaces of all three sets are the same. However, this assumption may limit the applicability of deep learning models because the target domain may have different data feature spaces, especially for unlabelled data (Gangopadhyay et al., 2020). Therefore, domain adaptation of deep learning models for cybersecurity is needed where data is scarce (Gangopadhyay et al., 2020). Domain adaptation is interchangeable with transfer learning.

Domain adaptation is considered a subset of transfer learning (Pan & Yang, 2010). The term is interchangeable with transductive transfer learning, defined to be the scenario where "the source and target tasks are the same, while the source and target domains are different." (Pan & Yang, 2010). In other words, using a trained model on a different but related domain can be considered domain adaptation. Two types of domain adaptation exist (Pan & Yang, 2010):

- The feature spaces between the source and target domains are different. Transfer learning between CPS and IoT is is an example of this.
- The feature spaces between domains are the same, but the marginal distribution of the input data is different.

Within the context of cybersecurity, transfer learning was designed to use knowledge from the source domain, which has sufficiently labelled data, to help build more precise

models in a related, but different, domain with only a few or no labelled data (Zhao, Shetty, Pan, Kamhoua & Kwiat, 2019). Then, the trained model would be refined using data from its actual domain. Because there are overlaps in attacks targeting IoT and CPS, certain datasets could be used to train (D)DoS detection models in both CPS and IoT, such as KDD99, CICIDS2017, ISCX2012 (R M Seyam et al., 2021; Ahmad & Alsmadi, 2021; Asharf et al., 2020).

According to (De, Bermudez-Edo, Xu & Cai, 2022), deep generative models have been applied for domain adaptation in the Industrial Internet of Things (IIoT). These include generative adversarial networks (GANs), autoregressive models (ARs), and variational autoencoder (VAE) (De et al., 2022; Tariq et al., 2020).

Additionally, we conducted a systematic search of existing studies for transfer learning and its subset domain adaptation on Google Scholar, ACM, IEEE, ScienceDirect, and Scopus databases. The results are presented in Fig. 6.1, which shows that there is currently a lack of research into domain adaptation for deep learning detection algorithms in cyber-physical systems. This presents an urgency to address this research gap. A sample search string that we used in our search is as follows.

"domain adaptation" AND "deep learning" AND (cyber physical system OR CPS OR internet-of-things OR IoT) AND (ddos OR dos OR denial-of-service)

Most existing research in transfer learning only focuses on either improving detection performance or solving the issue of imbalanced data in a chosen domain (T. Yang, Hou, Liu, Zhai & Niu, 2021; Haddaji, Ayed & Fourati, 2023). Few studies have looked into using transfer learning for cross-domain detection. Particularly, (Chen et al., 2023) proposed a cross-domain detection model that only considers transfer learning between SCADA and ICS networks. Other research uses weights of pre-trained, public CNN architectures for transfer learning rather than a specialised model for a chosen domain,

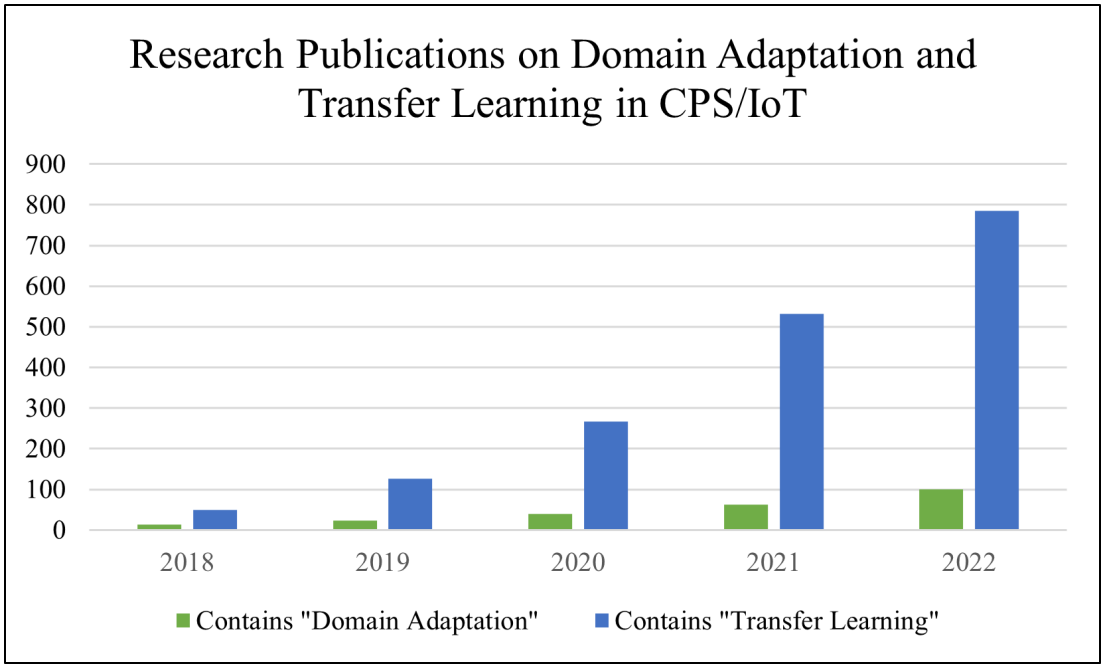

Figure 6.1: Research publications in domain adaptation in CPS/ICS domains

as seen in (Mehedi, Anwar, Rahman & Ahmed, 2021; Wang et al., 2021; Ben Atitallah, Driss, Boulila & Almomani, 2022).

6.3 Methodology

The purpose of this study is to test the capabilities of existing models (LUCID and MAD-GAN) for domain adaptation, and their learning capacities on different datasets. We only made minor changes to the source code as necessary to adapt them in our test environments, while ensuring that such changes do not affect the model architecture. We chose one (D)DoS detection model and a dataset for each detection domain (IoT and CPS), which resulted in two detection models and two datasets. We trained and evaluated each model's performance on each dataset to evaluate its capabilities for domain adaptation. Further details on the experiment design, models and dataset selection rationales are described below, with implementation details discussed in Section 6.4.

6.3.1 Models Selection

There are four rationales behind our model choices, being LUCID and MAD-GAN. These criteria ensure that the chosen models are credible, relevant to the context of the research, and accessible either publicly or through private sharing:

- **Context-relevance.** Section 6.2 has shown that RNN, LSTM, CNN, and autoencoder are popular for (D)DoS detection, while GAN is often used in domain adaptation. Therefore, the model must be one of these aforementioned types, or a similar hybrid model.
- **Domain-relevance.** The models must be applicable for detection in either CPS or IoT domains.
- **Credibility.** The model implementation must be backed by a well-cited research paper. Models published or last updated between 2019 and 2023 were prioritised since older models might be outdated. This does not exclude older models from consideration.
- **Availability.** The model source code must be available to reproduce its performance.

Using these search criteria, we have chosen the following two models: LUCID and MAD-GAN. We also considered various alternatives, including the 26 state-of-the-art models reported in a recent mapping study (Ngo, Mohaghegh & Sinha, 2022). However, these models were either not available upon request and/or not as well cited, despite reporting outstanding results.

LUCID is a CNN-based (D)DoS detection model that can be deployed in online resource-constrained environments, which had been tested in edge computing (Doriguzzi-Corin et al., 2020). The LUCID architecture comprises of a novel data preprocessing algorithm and the LUCID algorithm itself. LUCID's main training objective is to

Table 6.1: Rationales for Choosing LUCID and MAD-GAN

Criteria	**LUCID**	**MAD-GAN**
Context-relevance	Use CNN which is common in deep learning detection	Use GAN which is common in domain adaptation
Domain-relevance	Applicable for detection in IoT	Developed for detection in CPS
Credibility	Well-cited, last updated June 24, 2022(Doriguzzi-Corin et al., 2022)	Well-cited, published in 2019(D. Li, Chen et al., 2019)
Availability	Publicly available	Publicly available
Other rationale	Well-developed and complete for end users. Anyone can easily deploy LUCID	The model is not constrained by specific feature space size, thus making it very flexible for domain adaptation.

minimise its cost function by iteratively updating all the model's weights and biases, also known as trainable parameters. To optimise accuracy, LUCID performs a grid search through a set of hyperparameters with F1 as the evaluation metric. The training continues indefinitely for each point in the grid until the loss does not decrease for a consecutive 25 times. The F1 score is then saved before the model moves on to the next point. In our study, We use the latest version of LUCID, updated on June 24, 2022 (Doriguzzi-Corin et al., 2022). At the time of publication, the LUCID model was trained on the CICIDS2017 dataset and evaluated across another two datasets namely ISCX2012, CSECIC2018, and a mix of all three datasets called UNB201X(Doriguzzi-Corin et al., 2020).

MAD-GAN is a GAN model that incorporated LSTM-RNN for (D)DoS detection in CPS (D. Li, Chen et al., 2019; D. Li, Ng., Chen & Goh, 2019). The MAD-GAN model consists of a generator and a discriminator as two LSTM-RNNs (Long Short Term Memory Recurrent Neural Networks). MAD-GAN trains the generator and discriminator on the CPS devices' numerical measurements under normal operation in an unsupervised manner. The generator learns to reconstruct data similar to the CPS network's normal behaviours, while the discriminator learns to classify between real

and generated data. Thus, the generator and discriminator train each other based on the discriminator's loss and the generator's loss.

The architecture of MAD-GAN offers two methods for anomaly detection (D. Li, Chen et al., 2019). The first method uses only the discriminator for direct anomaly detection, while the second method uses both the generator and discriminator for detection (D. Li, Chen et al., 2019). However, we noticed that the second detection method could not function as expected. Similar issues were previously raised in 2019, but no solutions were found (D. Li, Ng. et al., 2019). As such, we have decided to only use the discriminator to evaluate both MAD-GAN's performance. This could affect MAD-GAN's detection performance.

6.3.2 Data Selection and Balancing

For this research, we prioritised using datasets supported by the two chosen models, which are CICIDS2017 (Sharafaldin. et al., 2018), CICDDoS2019 (of New Brunswick, n.d.; Sharafaldin, Lashkari, Hakak & Ghorbani, 2019b) and SWaT (*iTrust Labs Dataset Info*, 2022; Goh, Adepu, Junejo & Mathur, 2017) datasets. CICDDoS2019 contains 13 common (D)DoS attacks targeting computer networks and IoT devices (Sharafaldin et al., 2019b). The traffic in both CICIDS2017 and CICDDoS2019 datasets were generated using profiles representing abstract behaviours of human interactions with the testbed, allowing for the generation of naturalistic benign background traffic (Sharafaldin. et al., 2018; Sharafaldin et al., 2019b). Both datasets are frequently used by deep learning researchers for (D)DoS detection in the IoT domain. Meanwhile, the SWaT dataset contains attacks targeting CPS, including false data injection attacks (Goh et al., 2017). The SWaT dataset was collected from a real water treatment testbed and, as such, is also commonly used for the development of deep learning models for attack detection in the CPS environment and relevant domains (Goh et al., 2017). We selected the latest data

collected in December 2019 for the SWaT datasets, including both network packets and physical measurements. Both datasets are sufficiently large for our experiment.

One disadvantage of the CICDDoS2019 dataset is that it contains 99.9% attack traffic and only 0.1% benign traffic (Ferrag, Shu, Djallel & Choo, 2021). The amount of benign traffic in the CICDDoS2019 dataset is not sufficient to train LUCID and MAD-GAN, especially since LUCID's design requires that it maintain an exact 1:1 ratio of benign and malicious training traffic. In other words, we can only train on a large amount of attack traffic if we have an equivalent amount of benign traffic to maintain this ratio. As such, we collected all benign traffic recorded on Monday, July 3rd, 2017 by the University of Newbrunswick from the CICIDS2017 dataset. Henceforth, we refer to the combined portions from both CICIDS2017 and CICDDoS2019 datasets as the **CICDDoS dataset**.

Note that there are several similarities between CICIDS2017 and CICDDoS2019 that made this combination possible. Firstly, both datasets were realistically generated using the same profiling technique (Sharafaldin. et al., 2018; Sharafaldin et al., 2019b). The feature spaces of both datasets are also the same. As combining CICIDS2017 and CICIDDoS2019 have been done before by (Doriguzzi-Corin et al., 2020) to test LUCID's performance, we consider this method to be suitable to address the lack of benign traffic previously mentioned.

In our study, we chose the 2019 version of the SWaT dataset, which contains historian exfiltration and sensor disruption attacks targeting SCADA and engineering workstations (*iTrust Labs Dataset Info*, 2022).

6.3.3 Experiment Design

The experiment design is composed of two phases. The first phase involves training the two models and evaluating how well each model learns on their given dataset(s).

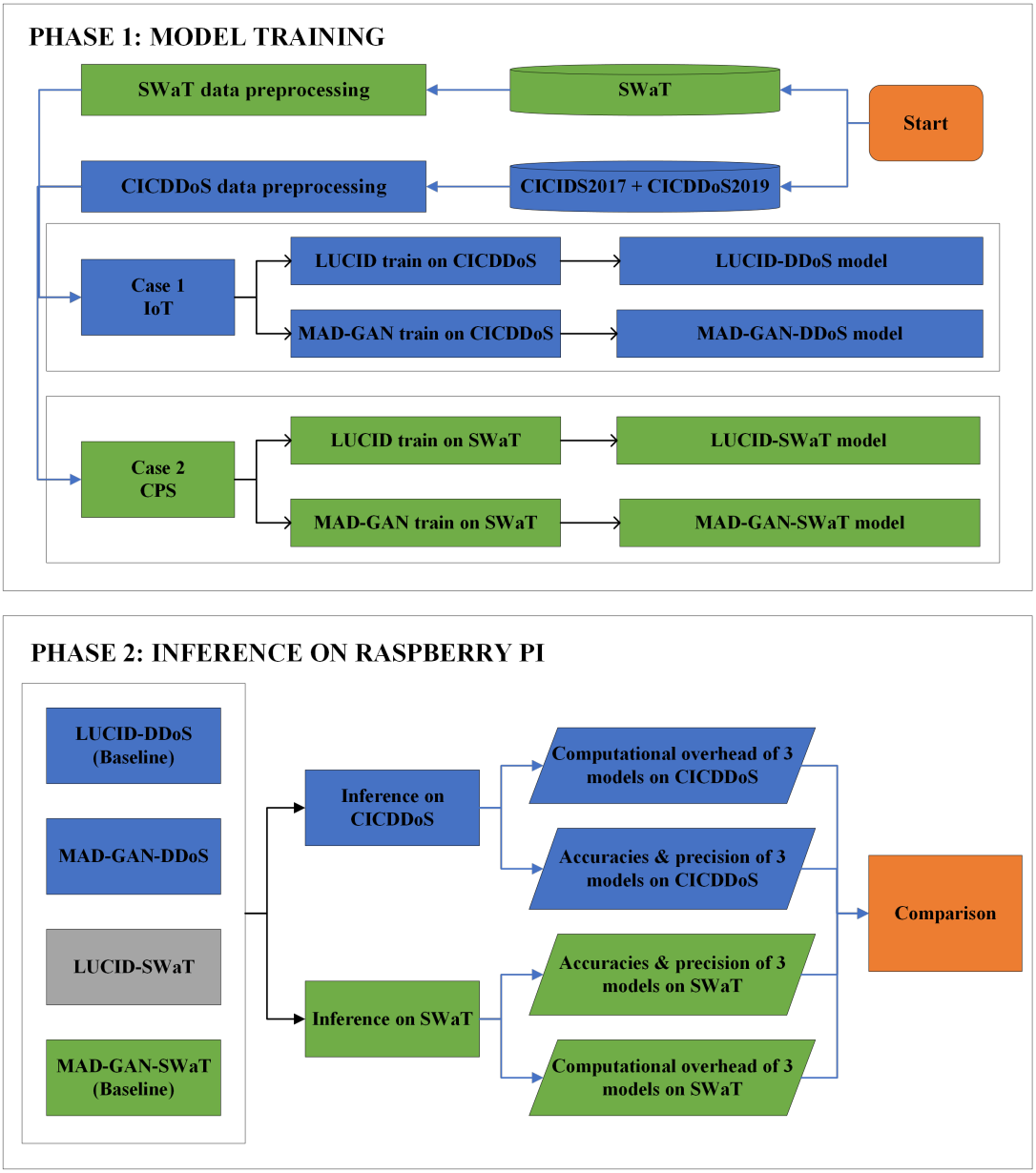

Figure 6.2: Experiment Design

The second phase involves testing the two models' detection performances and computational overhead in a resource-constrained environment with a Raspberry Pi. This is better represented in Fig. 6.2.

During both phase 1 and phase 2, we used the metrics of accuracy (Acc), precision

(Pre), recall (Rec), and F1 scores (F1) to compare the models' performances. These metrics are defined as follows.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.1}$$

$$Pre = \frac{TP}{TP + FP} \tag{6.2}$$

$$Rec = \frac{TP}{TP + FN} \tag{6.3}$$

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec} \tag{6.4}$$

Phase 1. The purpose of phase 1 is to train MAD-GAN and LUCID as part of our experiment on domain adaptation, as well as evaluate how well the models would adapt to unfamiliar datasets with different feature spaces.

We first perform data cleaning on the CICDDoS and SWaT datasets, as well as selecting data samples. We then propose two training cases for domain adaptation of the two models: IoT and CPS. Each case seeks to evaluate MAD-GAN and LUCID learning capability in unfamiliar domains. The first training case involves training both models on the CICDDoS datasets, representative of detection models for IoT environments. We refer to these two models as LUCID-DDoS and MAD-GAN-DDoS throughout the rest of this study. The second training case involves training LUCID and MAD-GAN on the SWaT dataset, which is representative of detection models for the CPS environment. Similar to case 1, the goal of case 2 is to also produce two models for (D)DoS attack detection in the CPS domain. We refer to these two models as LUCID-SWaT and MAD-GAN-SWaT throughout the rest of this study, respectively.

Phase 2. We expected to have four trained models by the time we reach phase 2 of the experiment, which are LUCID-DDoS, MAD-GAN-DDoS, LUCID-SWaT, and MAD-GAN-SWaT. In phase 2, we would test each model on the Raspberry Pi with

pre-recorded traffic traces from both CPS and IoT domains. Our aim is to evaluate the extent that domain adaptation affected each model's detection performance and computational overhead in a resource-constrained environment.

We chose a Raspberry Pi 4 to model a resource-constrained environment by limiting the CPU and RAM available to that of a low-end computer while providing the bare minimum technical requirement to run deep learning detection models. Using a physical Pi would also provide more values to our experiment results compared to using a virtual machine.

During our training with LUCID in phase 1, we discovered that LUCID was not able to learn meaningful information from the SWaT dataset, despite having faced no technical difficulties during training. We discussed such further details in Section 6.4, and how this development is related to our hypothesis. To reflect this, the experiment design in Fig. 6.2 shows a greyed-out LUCID-SWaT model. Therefore, only the three models, LUCID-DDoS, MAD-GAN-DDoS, and MAD-GAN-SWaT were deployed on the Raspberry Pi for evaluation.

In phase 2, we use MAD-GAN-SWaT's performance as the baseline for comparison When testing the three models MAD-GAN-SWaT, MAD-GAN-DDoS, and LUCID-DDoS on the SWaT dataset. Similarly, when the three models are tested on the CICDDoS dataset, LUCID-DDoS's performance would be the baseline instead. Accuracy, precision, recall, and F1 score are our performance metrics in this experiment. We would also measure the time taken to perform the detection task and the computational overhead of the models, defined as the CPU and Random Access Memory (RAM) usage of each model when performing detection on the Raspberry Pi.

6.4 Implementation

6.4.1 Hardware and Software

We used an Ubuntu 22.04 desktop in our training process for phase 1, with 1 TB hard drive storage and 128GB RAM and a Xeon W-2265 3.5 12C 24T 3.5 4.8 GHZ 19.25 MB cache processor. For the second phase, we used a Raspberry Pi 4 Model B Rev 1.4 with 7.6GB RAM to evaluate the models' detection performance.

We used two separate virtual environments to train the models, as each model used different package dependencies. Further details on dependencies used for training are in our GitHub repository[1].

6.4.2 Phase 1 - Case 1: (D)DoS in IoT

In case 1, we trained and tested both the LUCID and MAD-GAN models on the CICDDoS datasets. Because LUCID was developed for network detection, its performance in this phase was the baseline for comparison. This case yielded two models trained for (D)DoS detection in IoT.

LUCID-DDoS. We trained LUCID on the CICDDOS dataset with some changes made to the training hyperparameters. These changes are intended to reduce memory consumption while maintaining a high F1 score, based on case studies with LUCID reported in (Doriguzzi-Corin et al., 2020). Further information can be found in our GitHub[1].

MAD-GAN-DDoS. The MAD-GAN model's training and testing procedures require two `.txt` files containing the training and testing parameters for any new dataset;

[1] **All technical details and relevant resources on our experiment can be found at https://github.com/vickyngo-code/Domain-Adaptation-of-MAD-GAN-and-LUCID-for-Cyberattack-Detection**

therefore, we created similar `.txt` files customised for the CICDDoS dataset to adapt the MAD-GAN model. This proved that MAD-GAN has sufficient flexibility in its source code, thus satisfying condition C2 in our hypothesis.

We trained MAD-GAN on benign samples in the CICDDoS dataset before evaluating it on malicious samples. During the data preprocessing processes, we removed non-numerical columns, as well as replaced "NaN" and "Infinity" values with 0 instead. The model was trained over 100 epochs as per the default settings.

6.4.3 Phase 1 - Case 2: (D)DoS in CPS

In case 2, we trained and tested both the LUCID and MAD-GAN models on the SWaT datasets. We chose data from December 2019, which included both benign and malicious traffic in `.csv` and `.pcap` formats with full payloads. The dataset can be obtained by contacting the authors (*iTrust Labs Dataset Info*, 2022).

LUCID-SWaT. LUCID can adapt to any dataset, given that the IP addresses of the attacker and victim machines are available (Doriguzzi-Corin et al., 2022). This information is crucial for LUCID's data preprocessing algorithm to label benign and attack traffic accurately before feature extraction and training (Doriguzzi-Corin et al., 2022). The model extracts 11 features that are closely related to the TCP/IP protocols (Doriguzzi-Corin et al., 2020). LUCID's flexibility means that it has satisfied condition C2 in our hypothesis.

However, we discovered that LUCID's data preprocessing algorithm was unsuitable with the SWaT dataset because it does not contain definite attackers' or victims' IP addresses. This is very common in false data injection attack scenarios. As a result, LUCID was not able to extract meaningful patterns from the SWaT dataset to learn from. Therefore, we expect that LUCID would not be able to train or perform detection efficiently for protocol-based attacks other than TCP/IP protocol. Furthermore, LUCID

is tightly implemented around the TCP/IP protocol and does not extract packet payloads as part of its algorithm (Doriguzzi-Corin et al., 2022). This means that LUCID would be ineffective against application layer-based attacks in the CPS domain, where packet payloads play an important role in distinguishing benign and malicious traffic.

To summarise, LUCID's inability to train on the SWaT dataset can be attributed to its feature extraction algorithm. Although the LUCID model's design violated condition C1 in our hypothesis, we expect that this development would still align with our hypothesis, as later proven in Section 6.6. Improving the LUCID model for detection in CPS through further modification of the source code is out of the scope of this study.

MAD-GAN-SWaT. We chose to use the SWaT 2019 dataset as it provides the latest data for training in comparison with the SWaT 2016 dataset. Furthermore, our trial experiments with the SWaT 2016 dataset show that MAD-GAN's performance does not differ greatly from MAD-GAN-SWaTs performance with the SWaT 2019 dataset. Therefore, it would be more beneficial to use more recent data, as older data might not be representative of current attacks targeting CPS and IoT.

MAD-GAN-SWaT. We trained the MAD-GAN model on the 2019 version of the SWaT dataset, which is smaller and has fewer attacks compared to the 2016 version. However, we noticed that the small sample sizes in the SWaT 2019 dataset led to negative values in various arrays, causing errors. We concluded that a larger sample size is needed.

To solve this problem, we padded the dataset by duplicating existing data from the SWaT 2019 dataset for both benign and malicious samples. In principle, this solution discarded any assumptions of independence in the data samples while possibly leading to model overfitting. Using parts of the SWAT 2016 dataset as padding is unsuitable, as explained below:

- The measurements in the SWaT 2016 dataset are inconsistent with the measurements recorded in the SWaT 2019 datasets, indicating different states of the testbed.

- The minimum amount of data needed as padding from the SWaT 2016 dataset is twice the size of the SWaT 2019 dataset. Using this padding solution does not align with our main objectives of training MAD-GAN on the SWaT 2019 dataset.

6.4.4 Phase 2: Models Evaluation on Raspberry Pi 4

Following the experiments performed in case 1 and case 2, we made copies of the 3 trained models, namely LUCID-DDoS, MAD-GAN-DDoS, and MAD-GAN-SWaT. Then, and evaluated their performance in a resource-constrained environment modelled on a Raspberry Pi 4. It should be noted that LUCID was running offline `.pcap` inference mode. While it is a better practice to test the models in real-time traffic inference, there were several hardware limitations in our experiments:

- The Raspberry Pi 4 does not have sufficient hardware capabilities to run the models in real-time mode.

- We did not have suitable CPS and IoT testbeds to generate normal and attack traffic realistically.

- While the LUCID model has an option for live inference, the MAD-GAN model did not feature any real-time detection function. Given the Raspberry Pi limitations, it is fairer to test variations of both models with pre-recorded traffic.

It is important to note that in `.pcap` files inference mode, LUCID's only output is the DDoS rates, which depend entirely on the data that LUCID trained on. It is possible for LUCID to mistakenly classify benign traffic as DDoS traffic. Therefore,

we evaluated LUCID-DDoS on datasets that contain either entirely benign traffic or entirely malicious traffic to better evaluate LUCID's accuracy and false positive rate.

Meanwhile, the MAD-GAN models were trained and tested on labelled samples stored in the `.csv` format, allowing the models to calculate their detection metrics by comparing their predictions to the correct labels. Hence, MAD-GAN can be evaluated on a dataset that includes both benign and DDoS traffic. On the other hand, it would be difficult to measure MAD-GAN's detection accuracy in a real-time detection environment where it is required to perform detection on unlabelled data.

To accommodate MAD-GAN and LUCID's requirements, we propose the following 8 test cases to be carried out on the Raspberry Pi. Note that the CICDDoS dataset includes both benign and attack traffic, combined from CICIDS2017 and CICDDoS2019 respectively. In this phase, the testing set for MAD-GAN would have more attack samples than benign samples.

- 1) LUCID-DDoS on CICIDS2017 (benign)
- 2) LUCID-DDoS on CICDDoS2019 (attack)
- 3) LUCID-DDoS on SWaT (benign)
- 4) LUCID-DDoS on SWaT (attack)
- 5) MAD-GAN-DDoS on CICDDoS
- 6) MAD-GAN-DDoS on SWaT
- 7) MAD-GAN-SWaT on CICDDoS
- 8) MAD-GAN-SWaT on SWaT

Note that we have decided not to report the amount of training and testing data used, as we deemed this information trivial. This is due to the large discrepancies in

the amount of data used, which could be attributed to data availability and the chosen models' data processing capabilities rather than the testing methodology itself. Our reasoning for this is as follows:

- There was fewer data available in the `.csv` format from the SWaT 2019 dataset for MAD-GAN to train and test on. Meanwhile, SWaT 2019 has significantly more network packets data in the `.pcap` for use with LUCID.
- The `.csv` file format is unable to store as much network packet data as the `.pcap`, given the same file size. Therefore MAD-GAN is also capable of processing less data than LUCID.
- LUCID's data processing algorithm is much more efficient than MAD-GAN's, which allows it to process more data. Due to time constraints, it was impractical to train MAD-GAN-DDoS on as much data as we have used for LUCID, even without considering data availability.

6.5 Training Results Analysis (Phase 1)

In this section, we evaluate LUCID and MAD-GAN learning capabilities when trained on traffic traces on different domains. The results of phase 1 are especially relevant to condition 1 in our hypothesis, which stated that a model must be able to extract domain-relevant features to train effectively in an unfamiliar domain.

MAD-GAN models. The training results of MAD-GAN models were evaluated in terms of the discriminator's loss and the generator's loss over each epoch they were trained on. Both models were trained over 100 epochs. The GAN model design dictates that as the discriminator's loss decreases, the generator's loss should also decrease, and vice versa, as both the discriminator and generator are training each other (D. Li, Chen

Table 6.2: MAD-GAN Models' Average Discriminator & Generator losses

	MAD-GAN-SWaT	**MAD-GAN-DDoS**
Discriminator loss	0.505115	0.010799
Generator loss	4.715256	9.491622

et al., 2019). This implies that the two losses should eventually converge toward an optimal number after considerable training time.

Fig. 6.3 shows the discriminators' and generators' losses for both MAD-GAN-DDoS and MAD-GAN-SWaT models during training. The MAD-GAN-DDoS showed a continuous diverging pattern, which suggests that the MAD-GAN-DDoS model's discriminator was able to discriminate data produced by the generator exceptionally well, while not providing sufficient feedback for the generator to improve its performance. This led to increasing generator loss over time, as shown in the average loss over 100 epochs of both MAD-GAN models in Fig. 6.2. Furthermore, it might also imply that the discriminator did not train very well either, as the generator's worsening performance might not provide helpful feedbacks for the discriminator to improve. Since training MAD-GAN on both the SWaT and CICDDoS dataset caused a similar issue, the fact that MAD-GAn was not well-trained could be attributed to its architecture rather than the datasets themselves.

Overall, the results have shown that the MAD-GAN model has more balanced training results when trained on the SWaT dataset compared to training on the CICD-DoS dataset. Due to the divergence in the generators' losses in both models and the possibility that the discriminators might not be well-trained, we expect that the detection performance of both models will be impacted. Nevertheless, this shows that the MAD-GAN model satisfied both conditions C1 and C2 in our hypothesis, as the model was able to train on an unfamiliar dataset from the IoT domain. We expect that MAD-GAN will be able to perform cross-domain detection in phase 2.

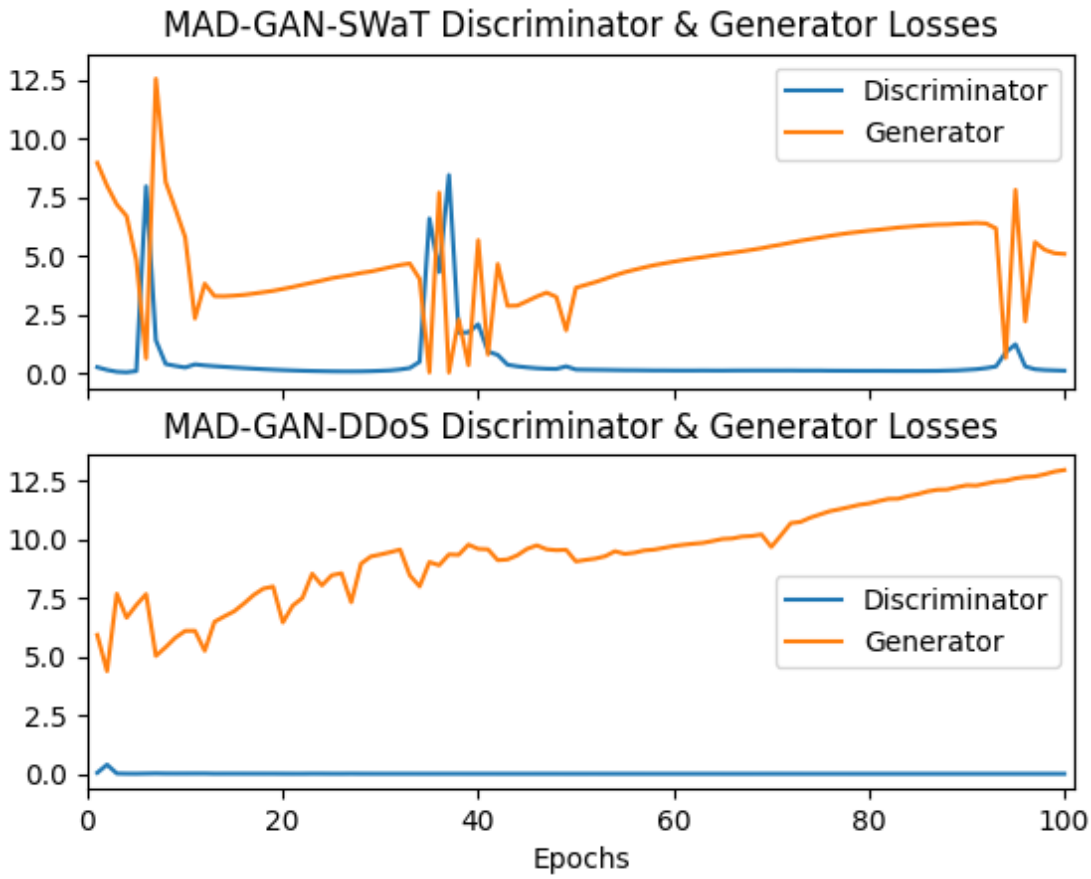

Figure 6.3: MAD-GAN Models Training Discriminator & Generator Loss

LUCID model. We extracted 872,590 samples, with 436295/436295 benign/DDoS distribution for training. The train/validation/test sizes are 706797/78534/87259 samples, which are equivalent to 5297712/583791/653940 packets, respectively. The LUCID-DDoS model achieved high detection performance on the training and testing sets, as shown in Table 6.3.

Because LUCID was not able to train on the SWaT dataset, we consider that it has violated condition C1 in our hypothesis. Despite LUCID having satisfied condition C2, this violation implies that LUCID should not be able to perform cross-domain detection, as per our hypothesis. In other words, we expect that LUCID-DDoS would not be able to discern between benign and malicious traffic in the SWaT dataset during phase 2.

6.6 Detection Performance Results Analysis (Phase 2)

We recorded LUCID-DDoS predicted DDoS percentage when evaluated on CICDDoS and SWaT datasets. On the other hand, MAD-GAN-DDoS and MAD-GAN-SWaT used the standard metrics, namely accuracy (Acc), precision (Pre), recall (Rec), and F1 scores, to evaluate its detection performance on the labelled data.

6.6.1 MAD-GAN Models

Both MAD-GAN-DDoS and MAD-GAN-SWaT were evaluated with attack data from the SWaT and CICDDoS datasets only, as the two models were already trained on benign data. For each dataset, we collected 10,000 samples (rows) for evaluation. For each test case, three types of epochs were used: sample-wise epoch, statistic-based

Table 6.3: LUCID-DDoS Training Results

	Accuracy	**F1 score**	**Precision**	**Recall**
Training	99.42%	99.43%	99.19%	99.67%
Testing	99.42%	99.42%	99.20%	99.63%

epoch, and logits-based epoch (D. Li, Ng. et al., 2019). The average results from the three epoch types for four test cases, calculated as the arithmetic mean, are reported in Table 6.4. Below, we compare both MAD-GAN-DDoS and MAD-GAN-SWaT performance on each dataset.

We observed that both MAD-GAN-SWaT and MAD-GAN-DDoS achieved near-perfect precision when tested on the CICDDoS dataset. On the same dataset, both accuracy and recall are the same regardless of the types of epochs. Low accuracy and recall also lowered the F1 score. These results suggest the following: 1) the number of false positives is close to zero, and 2) there might be many false negative predictions. Accuracy is the same as recall also implies that the number of true negatives must also be approximately zero. It is also possible that the CICDDoS dataset was imbalanced, leading to high numbers of false negatives yet an abysmal number of true negative predictions. The fact that both models did not train very well during phase 1 might also be an influencing factor.

When tested on the SWaT dataset, both models returned higher accuracy, but significantly lower precision when compared with previous testing results on the CICDDoS2019 dataset. The recall rates of MAD-GAN-SWaT and MAD-GAN-DDoS for all types of epochs do not exceed 50.35%. The F1 score is also low. Again, it is likely that the high number of false positives caused the low precision shown in Table 6.4. Interestingly, MAD-GAN-DDoS average performance on the SWaT dataset was on par with MAD-GAN-SWaT performance on the same dataset.

Table 6.4: MAD-GAN Models Evaluation Results

Test Case	**ACC**	**F1 score**	**PRE**	**REC**	**Time (s)**
MAD-GAN-SWaT on SWaT	55.39%	28.77%	37.09%	27.43%	1862
MAD-GAN-DDoS on SWaT	53.91%	28.53%	42.24%	30.26%	1955
MAD-GAN-SWaT on CICDDoS	35.98%	97.56%	35.98%	46.35%	1931
MAD-GAN-DDoS on CICDDoS	40.16%	98.83%	40.16%	50.60%	1931

The MAD-GAN models require extensive time to perform anomaly detection. Table 6.4 shows that each model took at least 30 minutes to classify 10,000 samples. Previous testing has shown a positive correlation between the amount of data and classification time, which could be a concern when deploying MAD-GAN for real-time detection, assuming that similar performance persists.

We recorded the CPU and RAM usage for MAD-GAN-SWaT and MAD-GAN-DDoS when evaluated on the SWaT and CICDDoS datasets, in that order, respectively. Fig. 6.4 shows that MAD-GAN-SWaT reached high CPU usage at the start, which eventually reduced to approximately 40%. Meanwhile, MAD-GAN-DDoS has a higher CPU usage overall. The fall in CPU usage to 0% signifies that the two models have finished the classification task, where there are minimal changes in RAM usage. This suggested that MAD-GAN requires extensive CPU power to perform detection at a reasonable speed, while low RAM usage means that the model is less likely to freeze the machine it was deployed on.

Overall, we have observed that the MAD-GAN model has the potential for domain adaptation and transfer learning, considering that MAD-GAN-DDoS achieved comparable performance with MAD-GAN-SWaT when evaluated on the SWaT dataset. However, it is questionable as to how well MAD-GAN would perform in a real-time detection environment on unlabelled due to the lack of evaluation technique and its slow detection time. Its high number of false negatives is also a concerning issue, although there are several factors affecting MAD-GAN's detection accuracies in our experiment. This includes MAD-GAN not being well-trained, using only the discriminator for detection, and overfitting issues from dataset padding as described in Section 6.4. On another hand, MAD-GAN's low RAM consumption would allow it to be deployed on a resource-constrained device. Future work is needed to improve MAD-GAN detection performance in a real-time, resource-constrained environment.

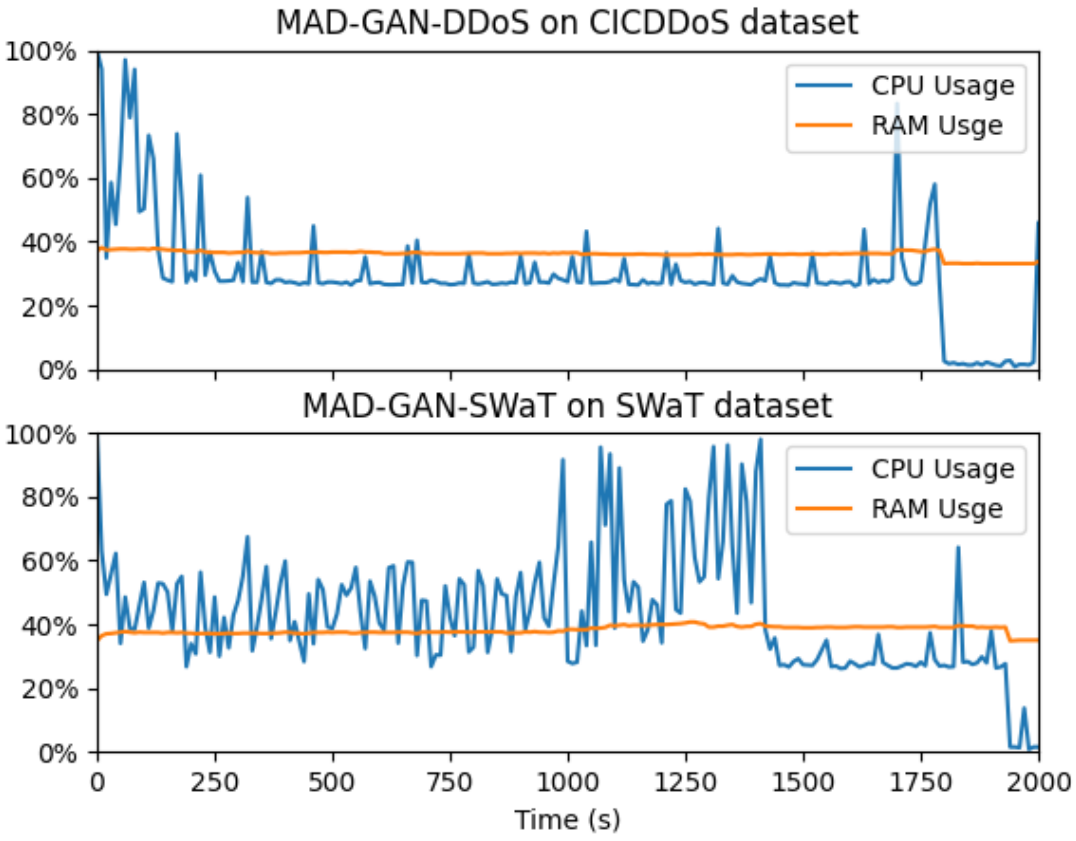

Figure 6.4: MAD-GAN Models CPU & RAM Consumption

Table 6.5: LUCID-DDoS Evaluation Results

Samples	**Packets**	**Time (s)**	**DDoS%**
cicddos_benign_1	44998	605.00	2.04%
cicddos_benign_2	42500	647.65	7.73%
cicddos_benign_3	34215	605.72	11.70%
cicddos_benign_4	25132	378.67	12.63%
cicddos_attack_1	26389	371.32	98.97%
cicddos_attack_2	26535	369.33	99.93%
cicddos_attack_3	26093	361.80	99.83%
cicddos_attack_4	25906	359.36	99.90%
cicddos_attack_5	25943	355.65	100.00%
cicddos_attack_6	25922	364.65	99.97%
swat_benign_1	77746	1007.91	33.22%
swat_benign_2	78190	998.26	34.34%
swat_attack_1	78879	1,010.46	33.79%
swat_attack_2	77137	986.00	30.82%

6.6.2 LUCID-DDoS

To determine LUCID-DDoS accuracy on unlabelled data, we evaluated LUCID-DDoS with fully benign and fully malicious data from both the SWaT and CICDDoS datasets. For each sample given to LUCID-DDoS, it will then perform detection in multiple instances, and return the DDoS rate as a percentage. Table. 6.5 reported the arithmetic mean of the DDoS rate for each instance. More information on the detected DDoS of each instance can be found in our GitHub repository [1].

The (D)DoS rate predicted by LUCID-DDoS on the CICDDoS datasets is shown in Table 6.5, where it reported near 100% DDoS rate in CICDDoS attack samples and lower DDoS rate in benign samples. This substantial difference suggested that LUCID-DDoS can effectively discern between benign and malicious traffic in the CICDDoS dataset. Nevertheless, LUCID-DDoS's detection results on benign sample 3 and 4 were much higher than the 0% DDoS rate that our initial expectations. We expect that LUCID-DDoS would to perform similarly when deployed in a real system for detection in IoT.

The LUCID (D)DoS percentage for two benign samples and two attack samples from the SWaT dataset is shown in Table 6.5. We found that LUCID had returned similar (D)DoS rates for both benign and traffic samples, which implies that LUCID-DDoS was not able to distinguish between benign and attack samples in the SWaT dataset. It also means that LUCID's training on the CICDDoS dataset did not help detect attacks in the CPS domain.

It is worth noting that the LUCID model only extracted that are tightly related to the TCP/IP protocol during both training detection (Doriguzzi-Corin et al., 2020). These features are vastly irrelevant to attacks that use different protocols other than TCP/IP. Furthermore, LUCID does not extract packet payload during detection, which made it ineffective against application-layer-based attacks in CPS. Further improvements are needed to adapt LUCID to a wider range of attack types and protocols.

Fig. 6.5 shows the CPU and RAM consumption from two test cases on the Raspberry Pi. It can be seen that LUCID has a high RAM usage demand for both test cases, while its CPU usage remains mostly stable at 30% except for some initial spikes. From Table 6.5, we also notice that samples with a higher number of packets tend to have a higher RAM consumption peak. This suggests that while LUCID-DDoS has a very low CPU usage, it should only perform anomaly detection on a low number of packets at a time to avoid maxing out the available RAM. In such cases, the machine it was deployed on could freeze and force a shutdown to resume operations. In ICS and CPS where availability requirements are high, such situations are not acceptable. Further work is needed to address LUCID-DDoS high RAM requirements.

To summarise, LUCID-DDoS is effective at detecting attacks that are strictly based on the TCP/IP protocol. While it was not able to classify attacks on the application layers, which include most industrial protocols as seen with the SWaT dataset, LUCID-DDoS has possible applications for detecting (D)DoS attacks in the IoT domain.

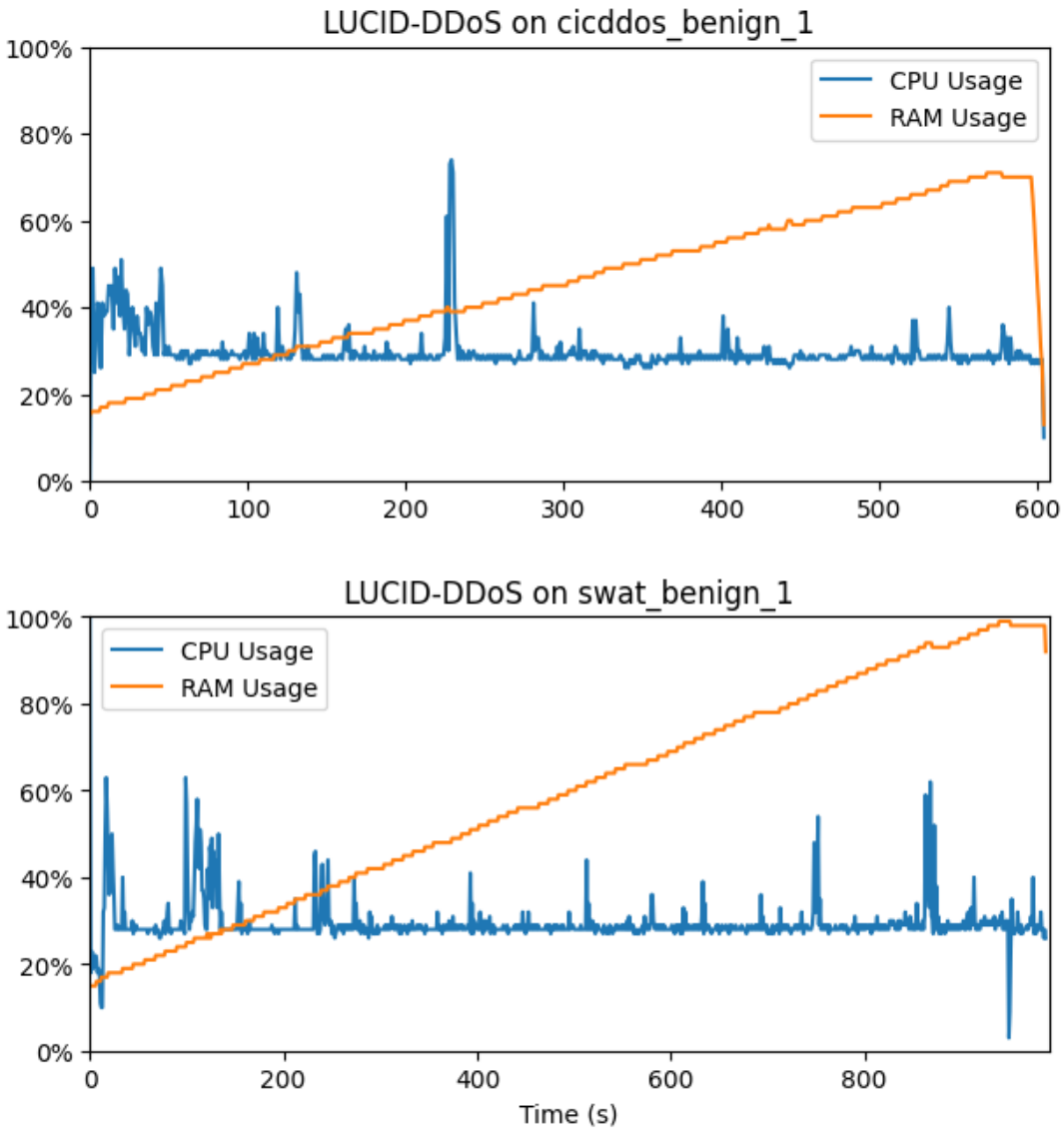

Figure 6.5: LUCID-DDoS CPU & RAM Consumption

6.6.3 Discussion & Threats to Validity

The experimental results have proven our hypothesis to be true, as shown by the fact MAD-GAN was able to perform cross-domain detection upon satisfying the two conditions described in Section 6.1. Because LUCID was not able to extract the relevant features from the SWaT dataset, it violated the condition C1 in our hypothesis for domain adaptation. As a result, it was unable to train on the SWaT dataset and performs cross-domain detection. This was a problem of architectural design rather than a technical difficulty.

LUCID-DDoS has significantly better performance than both MAD-GAN-SWaT and MAD-GAN-DDoS when evaluated on the CICDDoS dataset. While the MAD-GAN-SWaT did not achieve a good detection performance on the SWaT dataset, we still consider it to be more effective than the LUCID-DDoS performance on the same dataset. This is because LUCID-DDoS was not able to distinguish between benign and malicious traffic in the SWaT dataset at all with the data patterns it learned from training on the CICDDoS dataset, which means that there is no basis to compare LUCID-DDoS against MAD-GAN-SWaT. There are no observable changes to the three detection models' computational overhead when used to detect attacks in a different dataset. Despite using the same number of samples, we can see that MAD-GAN-SWaT has higher CPU usage than MAD-GAN-DDoS.

Furthermore, our experiments have confirmed that a model does not necessarily perform well on testing sets from unfamiliar domains. Particularly, we highlight the following points of consideration concerning domain adaptation and transfer learning in cybersecurity for CPS and IoT:

- Model architecture that allows for the extraction of relevant features for cross-domain training and detection.

- Maintaining low computational overhead when performing detection in a resource-constrained environment to avoid interruptions to concurrently running processes, or in severe cases, forced shutdown of the systems.

- Transferability of knowledge learned by detection models to ensure high detection accuracy when the model is deployed in a cross-domain detection environment.

There are also various threats to the validity of the experiment that must be addressed. Future work to address these limitations is mentioned briefly in Section 6.7. Keys threats to validity are noted as follows:

- **Differences in SWaT Dataset Training Data Used in LUCID and MAD-GAN models.** As we followed the same training methods and data format used for LUCID and MAD-GAN as described in their respective papers, there are some differences of note. Particularly, LUCID was trained on network packets from the SWaT dataset, while MAD-GAN was trained on the changes in physical characteristics of the system. We also used the newest data available, instead of those used in the original papers.

- **Lack of Testing Comparison with State-of-the-art.** In our experiment, we have only evaluated LUCID and MAD-GAN performance and not other state-of-the-art models. This is largely due to the lack of available resources to test all these models.

- **Lack of Resources for Real-time Detection Evaluation** As we lack a testbed that can generate realistic traffic to evaluate each model for real-time detection, we were only able to evaluate each model on recorded traffic. In contrast to LUCID, MAD-GAN does not have a real-time detection module, which means that further work is required to extend MAD-GAN capabilities before real-time evaluation.

6.7 Conclusion & Future Works

Our experiments have confirmed our hypothesis on domain adaptation for (D)DoS attack detection models for the IoT and CPS. Particularly, we can draw the following conclusions regarding domain adaptation within the context of detection in a resource-constrained environment:

- A model's detection accuracy following domain adaptation relies heavily on the model's ability to learn from relevant features and whether the patterns learned are transferrable to a different detection domain.
- Performing detection in an unfamiliar domain can result in poor detection results, as the learned data patterns may not transfer very well into the new domain. However, there are possible common points in (D)DoS attacks in the two domains of IoT and CPS that detection models could train on and use for domain adaptation.
- Domain adaptation does not have a large effect on a detection model's computational overhead, as it largely depends on the model's architectural design. Nevertheless, a model's computational overhead should also be given due consideration when deployed in a resource-constrained environment.

Further experiments should be performed in the future to test the validity of our hypothesis on other domains. Another research direction would be to explore possible solutions to address existing challenges in domain adaptation of models for deployment in resource-constrained environments.

Chapter 7

Discussion and Conclusions

This chapter provides a synthesis of the contributions of this research. We examine the existing machine learning landscape for use in the CPS domain and discuss how our findings shape the research direction toward domain adaptation of machine learning detection models in manuscript 2. Furthermore, we evaluate the implications and significance of our findings to the research field, as well as future research directions.

7.1 Discussion

We have identified from our SMS in manuscript 1 (Chapter 6) that existing datasets currently do not cover all (D)DoS attack types in the CPS domain, especially for protocol-based attacks. Due to existing models' high reliance on public datasets, it follows that current models' attack coverage could only be as large as their datasets' attack coverage. These findings emphasize a lack of detection models and datasets to address other (D)DoS attack methods in the CPS domain, which are practically unknown to these models. This is a serious, ongoing concern as the current research landscape prioritises improving detection accuracy and performance rather than expanding models' attack coverage (Ngo, Zahid et al., 2022). Inherently, there are three future approaches

to address (D)DoS attacks that have not been covered in the CPS domain: 1) develop new datasets to train existing or new models, 2) develop new detection models using training data from testbed of unaddressed attack types, and 3) expand existing models' attack coverage through domain adaptation. This is our first important finding.

Our second finding is that existing models give little consideration to the real-time deployment prospect of machine learning models. Although state-of-the-art models have addressed several relevant characteristics for actual deployment such as lightweight, scalability, and detection performance, there is still a lack of further testing of such models' performance in realistic, resource-constrained testbeds. Thus, it is unknown if we can efficiently apply these models to the industry context.

The two findings above help shape our future research direction toward finding a detection model that is capable of addressing the current low attack coverage of existing models, while also being suitable for deployment in a resource-constrained environment. By definition, a model that is capable of domain adaptation would also be able to adapt to different data feature spaces in the target domain (Pan & Yang, 2010) - a common phenomenon in actual model deployment. Because both issues are urgent and of high priority in the industry context, a practical solution, such as domain adaptation, would be needed. Once the model is successfully deployed on the CPS systems, it can gradually adapt and learn from the incoming traffic that it received. In other words, this approach uses transfer learning to form an existing model's foundational knowledge before letting it improves its detection capabilities through real-time traffic in its environment. On the other hand, the development of new models and datasets would surely take considerable time and effort. Compared to the first two methods, using domain adaptation is both resource-efficient and time efficient in the industry context. Our research aims to evaluate the feasibility of domain adaptation as a solution to the aforementioned issue.

It should be noted that there are other alternatives to our approach in addressing the lack of attack coverage in existing datasets and detection models in the CPS domain.

The most straightforward solution would still be to develop a testbed to collect traffic on previously unaddressed attacks, which can be trained by existing or new detection models, as we have mentioned previously. In such cases, the testbed design would play a crucial role in generating realistic traffic that can represent the actual attacks in the industry. To the best of our knowledge, there are unfortunately no standard frameworks concerning the development of testbeds for dataset generation purposes, evaluating dataset quality a challenging task.

Interestingly, publications reporting new detection models are abundant in top publishing venues as per Fig. 3.3. However, there are lesser research focus on the development of datasets and the usage of domain adaptation for cybersecurity as shown in Fig. 6.1. Our background research in Chapter 6 shows that little is known about what criteria would facilitate domain adaptation in a detection model. Therefore, it is important for us to first establish the minimum requirements for the domain adaptation to be feasible. Because domain adaptation inherently requires the model to transfer and apply its knowledge in an unfamiliar detection domain, we proposed a hypothesis on these requirements, as follows.

A deep learning (D)DoS detection model is capable of domain adaptation given that it satisfies the following two conditions.

- **C1 - Transferability of Learned Knowledge:** The model can extract domain-relevant features for training and performing detection.
- **C2 - Technical Flexibility:** The detection model source code must allow for alteration in such a way that the model can be easily adapted to datasets from different domains without major changes in its architecture. Thus, hard-coded values that are strictly specific to certain datasets are discouraged.

We considered the controlled experiment approach to be the best to test our hypothesis, as discussed in Chapter 4.

The controlled experiment is discussed in manuscript 2 (Chapter 6). Although we were successful in establishing C1 and C2 as the minimum requirements for domain adaptation to be possible, many other implications for the detection model's performance limit its applicability for real-time detection. Firstly, we should note that the "knowledge transfer" aspect of condition C1 prioritises detection accuracy and performance, while the "technical flexibility" aspect of condition C2 prioritises efficient model architecture for adapting to different data feature spaces. On another hand, the context of "deployment in resource-constrained environments" values low computational overhead and the quality of being lightweight the most. For a model to perform domain adaptation successfully while returning high detection results, all three aspects must be considered, including their respective priorities. To focus on one priority or characteristic would be to focus on that certain aspect. For each of these aspects and their priorities, the experiment results have shown:

- **Knowledge Transfer**: A detection model is highly reliant on its feature extraction module design to satisfy condition C1. This module also plays a role in determining a model's detection accuracy and performance in an unfamiliar detection domain, depending on whether the features extracted are indeed relevant to the target domain. Detection accuracy could be said to be most heavily impacted in domain adaptation.

- **Technical flexibility**: A model's architecture ultimately determines the data distribution and feature spaces that a detection model can work with. Fundamentally, technical flexibility would be the first entry gate in deciding whether a model can perform domain adaptation.

- **Deployment in resource-constrained environments**: Again, a model's architecture inherently determines how computationally intensive a model can be during training, and most importantly, during detection. While it is possible to optimise

Table 7.1: Framework for Domain Adaptation of Deep Learning Detection Models in CPS Domain

Aspects	**Optimisation Priorities**	**Influencing Components**	**Recommendations**
Knowledge Transfer	Learning capabilities of cross-domain features, detection accuracy and performance	Feature extraction algorithm, model training module	Maximise extracting and learning of features that are relevant to cross-domain detection.
Technical Flexibility	Adaptation to different data feature spaces, and there by different datasets	Model architecture and design	Minimise fixed feature space requirements during model training and detection phase
Deployment in Resource-Constrained Environments	Low computational overhead (CPU & RAM), lightweight, detection speed, real-time detection	Model architecture	Incorporation of relevant technical optimisations on the model architecture during and after the design stage.

a model's computational overhead at a later stage, it would be more efficient to consider these priorities during the model design stage.

These findings can be consolidated into a framework for domain adaptation of deep learning detection models in CPS domain, as follows.

Although the three aspects of domain adaptations seem to have vastly different optimisation priorities, they all involve the model's architecture as the highest tier for optimisation. Future researchers on domain adaptation should take into consideration all three aspects during the model design stage, given how important these designs would affect the model's detection capabilities during later stages. As examples of using this framework for future research directions in domain adaptation for cybersecurity, we recommend researchers consider optimising the model's feature extraction algorithm

such that the model could extract features that are relevant to both the source and target domains, which would maximise detection accuracy. We suggest that such a model be designed to accept raw traffic packets as they would include all features and characteristics, and thus are more informative. Raw traffic packets are most commonly recorded in `.pcap` format. To achieve low computational overhead, we propose researchers reduce heavy computations and balance both RAM and CPU usage. This is because high RAM usage could lead to the system's freezing, which is often not acceptable in the CPS environment where system availability is most important. In serious cases, the systems could be forced into a restart.

To summarise, we established the minimum requirements for domain adaptation of detection models to be feasible, thereby demonstrating that domain adaptation is a realistic solution to the existing low coverage of (D)DoS attack types in the CPS domain. We also discussed the three aspects needed for domain adaptation to be usable in the real industry as a framework, and demonstrate the effect of domain adaptation on a detection model's performance.

7.2 Threats to Validity

We evaluate the threats to validity of our research according to (Wohlin et al., 2012), which are construct validity and external validity

7.2.1 Construct Validity

Assumption of Completeness

Most machine learning detection models in Chapter 3 do not mention the specific attack types used in each dataset. Therefore, we assume that the model was trained on all attacks in any given dataset, unless mentioned otherwise. This implies that the actual

attack coverage of existing detection models might be smaller than reported.

Technical Issues during Domain Adaptation of the MAD-GAN Model

There were various technical issues with MAD-GAN that we had to circumvent during our experiment, which might have had a major impact on its detection performance. Firstly, MAD-GAN-SWaT was likely to have been overfitted due to the training dataset padding that we performed in Section 6.4. Secondly, MAD-GAN-SWaT and MAD-GAN-DDoS were not able to use both the discriminator and generator for attack detection due to technical difficulties, which might have been able to return better detection performance.

Differences in Training Data in LUCID and MAD-GAN models

Due to how MAD-GAN and LUCID were designed, there are some differences in the SWaT training data used for both models. Specifically, LUCID was only able to train on network traffic in the SWaT dataset. Meanwhile, the MAD-GAN model was trained on the physical characteristics of the SWaT testbed during normal operations rather than network packages. Despite the discrepancy, we still decided to train MAD-GAN on physical characteristics because we would like to assess MAD-GAN's potential at the time of publication using the same training methods described in (D. Li, Chen et al., 2019).

7.2.2 External Validity

Exclusion of detection models from other domains

Although some detection models in different domain (e.g. computer network) can be applied into the CPS context through domain adaptation, we did not have available resources to verify each model for its applicability in ICS-relevant domains. As such, our

research only considers machine learning and deep learning (D)DoS detection models from CPS-relevant domains, as they were already developed for (D)DoS detection in that environment.

Exclusion of studies on optimisation techniques

Since manuscript 1 (Chapter 3) focuses on mapping existing state-of-the-art machine learning (D)DoS detection models, we consider studies that primarily focus on optimisation techniques for machine learning detection model to be out of scope. However, we do consider elements that existing models have optimized, as shown in Table 3.8.

Lack of Experimentations on other State-of-the-art Models

In our research, we have only evaluated LUCID and MAD-GAN detection performance but not other state-of-the-art models because they do not satisfy our selection criteria in Table 6.1, especially the availability criteria. Further experiments are needed to strengthen the validity of our hypothesis.

Lack of Real-time Detection Performance Evaluation on Resource-Constrained Testbed

Due to resource limitations, we were not able to develop a realistic resource-constrained testbed to evaluate LUCID and MAD-GAN in a real-time detection context. Although real-time detection was not a part of our hypothesis, it is still a crucial element in considering the applications of domain adaptation in industry systems.

7.3 Conclusions

We demonstrated that domain adaptation of MAD-GAN and LUCID is a realistic solution to address the low attack coverage of existing (D)DoS attack methods. It is

the most efficient solution in terms of time and resources and is also practical in that it solves the issue with different data feature spaces in real-time deployment. Additionally, we have also established conditions C1 and C2 of our hypothesis to be the minimum requirements for domain adaptation of detection models to be feasible. As mentioned in manuscript 2 (Chapter 6), there are two possible outcomes that would support our hypothesis. The MAD-GAN and LUCID domain adaptation experiment results demonstrated both of these outcomes.

We also demonstrated how a model's detection performance can be improved in domain adaptation. We identified three important aspects of domain adaptation that make it a practical solution, which are the "knowledge transfer" aspect, the "technical flexibility" aspect, and the "deployment in resource-constrained environments" aspect. Each aspect value different priorities, which can be directly influenced by different underlying factors. However, these underlying factors all pointed toward one central component: model architecture. To put it simply, the model's architecture and design play an extremely important role in domain adaptation as it determines the technical flexibility of the model to adapt to different feature spaces, which domain-relevant feature to extract to maximise learning and detection performance, and how computationally intensive the models would be when performing detection. Given the large influence that a model's architecture has on its detection, and performance, it would perhaps be more efficient for future researchers to consider the three aspects of domain adaptation during the model architectural design stage rather than during the model's completion.

Essentially, our research provided a framework through which future researchers would be able to form the baseline considerations in designing a detection model that is optimised for transfer learning and cross-domain detection based on the three aforementioned aspects. We have also made several recommendations for the future development of detection models for domain adaptation in Table. 7.1. We suggest that

future research consider the optimisation of machine learning models in a resource-constrained environment context, where qualities that make a model applicable for deployment in the industry are also prioritised along with detection accuracy.

7.4 Future Works

As discussed in Chapter 3, the attack coverage of existing datasets has a high impact on machine learning detection models' attack coverage. We have identified that most studies relied heavily on datasets in addition to testbeds to train their models. Therefore, we consider the development of new datasets to cover for currently unaddressed (D)DoS attack types through data collection on a realistic testbed. Additionally, we also consider developing novel machine learning models that address certain attack types that have not been addressed thus far. For this research direction, a realistic testbed to collect data from is required, as there currently exist no suitable datasets for this task. Furthermore, we also intend to work on the optimisation of these models for real-time deployment in a resource-constrained environment, which are common in real ICS in the industry.

This leads us to another important research direction on testbeds for ICS cybersecurity. Through our mapping study, we have identified that there lacks a framework or standard for the development of a realistic testbed that resembles real-world systems. Such a framework can be developed through case studies of real ICS in the industry. Through these case studies, it is possible to consolidate a set of minimum requirements and essential components constituting an ICS network.

On another hand, we have also identified that domain adaptation for cybersecurity is a relatively new research area. Domain adaptation has had extensive applicability in other research areas, such as computer vision and natural language processing. Domain adaptation implies that a machine learning model can utilise learned patterns from a relevant detection task to a different detection domain. Therefore, it is a promising

solution to the lack of datasets for training machine learning (D)DoS detection models in CPS. Nevertheless, there are still many unknowns in using domain adaptation for cybersecurity.

One future research direction would be to perform extensive testing and experiments on domain adaptation of other detection models on different attack types. This will help us strengthen our hypothesis and identify any additional factors in domain adaptation that we might not have been able to observe in our past works. It would also be worth performing additional experiments to test our hypothesis in other domains.

We intend to work on a novel cross-domain deep learning (D)DoS detection model for CPS and IoT as the two domains are related. Particularly, we seek to identify the most optimal design to facilitate domain adaptation or (D)DoS detection in both CPS and IoT, and potentially other domains such as computer networks. At the same time, we also aim to optimise the model's performance for real-time deployment in resource-constrained environments.

References

Alexander Gutnikov, Oleg Kupreev & Yaroslav Shmelev. (2022, 8). *DDoS attacks in Q2 2022* (Tech. Rep.). Retrieved from https://securelist.com/ddos-attacks-in-q2-2022/107025/

Abdelaty, M., Doriguzzi-Corin, R. & Siracusa, D. (2020). AADS: A Noise-Robust Anomaly Detection Framework for Industrial Control Systems. In J. Zhou, X. Luo, Q. Shen & Z. Xu (Eds.), *Information and Communications Security* (pp. 53–70). Cham: Springer International Publishing.

Ahmad, R. & Alsmadi, I. (2021). Machine learning approaches to IoT security: A systematic literature review. *Internet of Things*, *14*, 100365. Retrieved from https://www.sciencedirect.com/science/article/pii/S2542660521000093 doi: https://doi.org/10.1016/j.iot.2021.100365

Alimi, O. A., Ouahada, K., Abu-Mahfouz, A. M., Rimer, S. & Alimi, K. O. A. (2021, 8). A review of research works on supervised learning algorithms for scada intrusion detection and classification. *Sustainability*, *13*, 9597. Retrieved from https://www.mdpi.com/2071-1050/13/17/9597 doi: 10.3390/su13179597

Arora, P., Kaur, B. & Teixeira, M. A. (2022). *Security in industrial control systems using machine learning algorithms: An overview* (Vol. 314). Springer Singapore. Retrieved from http://dx.doi.org/10.1007/978-981-16-5655-2_34 doi: 10.1007/978-981-16-5655-2_34

Artificial Intelligence Solutions. (2022). Retrieved from https://www.sparkcognition.com/

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W. & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics*, *9*(7). Retrieved from https://www.mdpi.com/2079-9292/9/7/1177 doi: 10.3390/electronics9071177

Babu, M. R. & Veena, K. N. (2021). A Survey on Attack Detection Methods For IoT Using Machine Learning And Deep Learning. In *2021 3rd international conference on signal processing and communication (icpsc)* (p. 625-630). doi: 10.1109/ICSPC51351.2021.9451740

Ben Atitallah, S., Driss, M., Boulila, W. & Almomani, I. (2022). An effective detection and classification approach for dos attacks in wireless sensor networks using deep

transfer learning models and majority voting. In C. Bădică, J. Treur, D. Benslimane, B. Hnatkowska & M. Krótkiewicz (Eds.), *Advances in computational collective intelligence* (pp. 180–192). Cham: Springer International Publishing.

Chan, R., Chow, K.-P. & Chan, C.-F. (2019). Defining Attack Patterns for Industrial Control Systems BT - Critical Infrastructure Protection XIII. In J. Staggs & S. Shenoi (Eds.), (pp. 289–309). Cham: Springer International Publishing.

Check Point Software Technologies Ltd. (2020). *Top 10 Critical Infrastructure and SCADA/ICS Cybersecurity vulnerabilities and threats* (Tech. Rep.). Retrieved from https://www.checkpoint.com/downloads/products/top-10-cybersecurity-vulnerabilities-threat-for-critical-infrastructure-scada-ics.pdf

Chen, Y., Su, S., Yu, D., He, H., Wang, X., Ma, Y. & Guo, H. (2023). Cross-domain industrial intrusion detection deep model trained with imbalanced data. *IEEE Internet of Things Journal*, *10*(1), 584-596. doi: 10.1109/JIOT.2022.3201888

Chhillar, S. (2020, 12). *Common ICS Cybersecurity Myth #1: The Air Gap.* Retrieved from https://gca.isa.org/blog/common-ics-cybersecurity-myth-1-the-air-gap

Communications Security Establishment (CSE) & Canadian Institute for Cybersecurity (CIC). (2018). *IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.* Retrieved from https://www.unb.ca/cic/datasets/ids-2018.html

De, S., Bermudez-Edo, M., Xu, H. & Cai, Z. (2022). Deep Generative Models in the Industrial Internet of Things: A Survey. *IEEE Transactions on Industrial Informatics*, *18*(9), 5728-5737. doi: 10.1109/TII.2022.3155656

Derhab, A., Guerroumi, M., Gumaei, A., Maglaras, L., Ferrag, M. A., Mukherjee, M. & Khan, F. A. (2019). Blockchain and random subspace learning-based ids for sdn-enabled industrial iot security. *Sensors*, *19*(14). Retrieved from https://www.mdpi.com/1424-8220/19/14/3119 doi: 10.3390/s19143119

Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-Del-Rincon, J. & Siracusa, D. (2020). Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection. *IEEE Transactions on Network and Service Management*, *17*, 876-889. doi: 10.1109/TNSM.2020.2971776

Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martínez-del Rincón, J. & Siracusa, D. (2022, June). *LUCID: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection.* https://github.com/doriguzzi/lucid-ddos.

Easterbrook, S., Singer, J., Storey, M.-A. & Damian, D. (2008). *Selecting Empirical Methods for Software Engineering Research.*

Farahani, A., Voghoei, S., Rasheed, K. & Arabnia, H. R. (2020). *A brief review of domain adaptation.* arXiv. Retrieved from https://arxiv.org/abs/2010.03978 doi: 10.48550/ARXIV.2010.03978

Ferrag, M. A. & Maglaras, L. (2020). Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids. *IEEE Transactions on Engineering Management*, *67*(4), 1285-1297. doi: 10.1109/TEM.2019.2922936

Ferrag, M. A., Shu, L., Djallel, H. & Choo, K.-K. R. (2021). Deep Learning-Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0. *Electronics*, *10*(11). Retrieved from https://www.mdpi.com/2079-9292/10/11/1257 doi: 10.3390/electronics10111257

Gangopadhyay, A., Odebode, I. & Yesha, Y. (2020). A domain adaptation technique for deep learning in cybersecurity. In (Vol. 11878 LNCS, p. 221-228). Springer. doi: 10.1007/978-3-030-40907-4_24

Goh, J., Adepu, S., Junejo, K. N. & Mathur, A. (2017). A Dataset to Support Research in the Design of Secure Water Treatment Systems. In G. Havarneanu, R. Setola, H. Nassopoulos & S. Wolthusen (Eds.), *Critical Information Infrastructures Security* (pp. 88–99). Cham: Springer International Publishing.

Guidelines for performing systematic literature reviews in software engineering (Tech. Rep. No. EBSE 2007-001). (2007, 09 07). Keele University and Durham University Joint Report. Retrieved from https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

Haddaji, A., Ayed, S. & Fourati, L. C. (2023). A transfer learning based intrusion detection system for internet of vehicles. In *2023 15th international conference on developments in esystems engineering (dese)* (p. 533-539). doi: 10.1109/DeSE58274.2023.10099623

Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). *Design Science in Information Systems Research* (Vol. 28). Retrieved from https://www.jstor.org/stable/25148625

Horak, T., Strelec, P., Huraj, L., Tanuska, P., Vaclavova, A. & Kebisek, M. (2021). The vulnerability of the production line using industrial iot systems under ddos attack. *Electronics*, *10*(4). Retrieved from https://www.mdpi.com/2079-9292/10/4/381 doi: 10.3390/electronics10040381

Huraj, L., Horak, T., Strelec, P. & Tanuska, P. (2021). Mitigation against ddos attacks on an iot-based production line using machine learning. *Applied Sciences*, *11*(4). Retrieved from https://www.mdpi.com/2076-3417/11/4/1847 doi: 10.3390/app11041847

Ibrahim, R. F., Abu Al-Haija, Q. & Ahmad, A. (2022). DDoS attack prevention for internet of thing devices using Ethereum Blockchain technology. *Sensors*, *22*(18), 6806. doi: 10.3390/s22186806

Industrial Intelligence. (2022). Retrieved from https://www.uptake.com/

iTrust Labs Dataset Info. (2022, May). Retrieved from https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/

Khaitan, S. K. & McCalley, J. D. (2015). Design Techniques and Applications of Cyberphysical Systems: A Survey. *IEEE Systems Journal*, *9*(2), 350-365. doi: 10.1109/JSYST.2014.2322503

Khan, M. A., Karim, M. R. & Kim, Y. (2019). A scalable and hybrid intrusion detection system based on the convolutional-lstm network. *Symmetry*, *11*(4). Retrieved from https://www.mdpi.com/2073-8994/11/4/583 doi: 10.3390/sym11040583

Kiran, D. (2019). Chapter 35 - Internet of Things. In D. Kiran

(Ed.), *Production planning and control* (p. 495-513). Butterworth-Heinemann. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780128183649000354 doi: https://doi.org/10.1016/B978-0-12-818364-9.00035-4

Knapp, E. D. & Langill, J. T. (2015). Chapter 2 - about industrial networks. In E. D. Knapp & J. T. Langill (Eds.), *Industrial network security (second edition)* (Second Edition ed., p. 9-40). Boston: Syngress. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780124201149000022 doi: https://doi.org/10.1016/B978-0-12-420114-9.00002-2

Kothari, C. R. (2004). *Research Methodology: Methods and Techniques*. New Age International.

Lee, E. A. & Seshia, S. A. (2016, Dec). *Introduction to Embedded Systems: A Cyber-Physical Systems Approach.* https://mitpress.mit.edu/9780262533812/introduction-to-embedded-systems/.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J. & Ng, S.-K. (2019). MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In *Artificial neural networks and machine learning – icann 2019: Text and time series: 28th international conference on artificial neural networks, munich, germany, september 17–19, 2019, proceedings, part iv* (p. 703–716). Berlin, Heidelberg: Springer-Verlag. Retrieved from https://doi.org/10.1007/978-3-030-30490-4_56 doi: 10.1007/978-3-030-30490-4_56

Li, D., Ng., S.-K., Chen, D. & Goh, J. (2019, Jan). *Multivariate Anomaly Detection for Time Series Data with GANs.* https://github.com/LiDan456/MAD-GANs.

Li, Y. & Wang, Y. (2020). Developing graphical detection techniques for maintaining state estimation integrity against false data injection attack in integrated electric cyber-physical system. *Journal of Systems Architecture*, *105*, 101705. Retrieved from https://www.sciencedirect.com/science/article/pii/S1383762119305120 doi: https://doi.org/10.1016/j.sysarc.2019.101705

Mehedi, S. T., Anwar, A., Rahman, Z. & Ahmed, K. (2021). Deep transfer learning based intrusion detection system for electric vehicular networks. *Sensors*, *21*(14). Retrieved from https://www.mdpi.com/1424-8220/21/14/4736 doi: 10.3390/s21144736

Mirkovic, J. & Reiher, P. (2004, apr). A taxonomy of ddos attack and ddos defense mechanisms. , *34*(2), 39–53. Retrieved from https://doi.org/10.1145/997150.997156 doi: 10.1145/997150.997156

Mittal, M., Kumar, K. & Behal, S. (2022). Deep learning approaches for detecting ddos attacks: a systematic review. *Soft Computing*. Retrieved from https://doi.org/10.1007/s00500-021-06608-1 doi: 10.1007/s00500-021-06608-1

Morris, T. (n.d.). *Industrial Control System (ICS) cyber attack datasets.* Retrieved from https://sites.google.com/a/uah.edu/

tommy-morris-uah/ics-data-sets

Mujeeb Ahmed, C., Umer, M. A., Binte Liyakkathali, B. S. S., Jilani, M. T. & Zhou, J. (2021). Machine Learning for CPS Security: Applications, Challenges and Recommendations. In Y. Maleh, M. Shojafar, M. Alazab & Y. Baddi (Eds.), *Machine Intelligence and Big Data Analytics for Cybersecurity Applications* (pp. 397–421). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-030-57024-8_18 doi: 10.1007/978-3-030-57024-8_18

NCCIC, I.-C. (2016). *Ics-cert annual assessment report industrial control systems cyber emergency response team.*

Ngo, V., Mohaghegh, M. & Sinha, R. (2022). *Domain-Adaptation-of-MAD-GAN-and-LUCID-for-Cyberattack-Detection.* Retrieved from https://github.com/vickyngo-code/Domain-Adaptation-of-MAD-GAN-and-LUCID-for-Cyberattack-Detection (https://github.com/vickyngo-code/Domain-Adaptation-of-MAD-GAN-and-LUCID-for-Cyberattack-Detection)

Ngo, V., Zahid, F., Mohaghegh, M. & Sinha, R. (2022). A Systematic Mapping of Datasets and Machine Learning Models for (D)DoS Detection in Industrial Control Systems. *IEEE Transactions of Neural Networks and Learning Systems*. (Under review)

of New Brunswick, U. (n.d.). *DDoS 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.* https://www.unb.ca/cic/datasets/ddos-2019.html.

Pan, S. J. & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359. doi: 10.1109/TKDE.2009.191

A review of intrusion detection systems: Datasets and machine learning methods. (2021). Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3454127.3456576 doi: 10.1145/3454127.3456576

R M Seyam, A., Bou Nassif, A., Nasir, Q., Al Blooshi, B. & Abu Talib, M. (2021). Deep learning techniques to detect dos attacks on industrial control systems: A systematic literature review. In *The 7th annual international conference on arab women in computing in conjunction with the 2nd forum of women in research.* New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3485557.3485577 doi: 10.1145/3485557.3485577

Salim, M. M., Rathore, S. & Park, J. H. (2020). Distributed denial of service attacks and its defenses in iot: a survey. *Journal of Supercomputing*, *76*, 5320-5363. Retrieved from https://doi.org/10.1007/s11227-019-02945-z doi: 10.1007/s11227-019-02945-z

Schneider, P. & Böttinger, K. (2018). High-performance unsupervised anomaly detection for cyber-physical system networks. In (p. 1–12). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3264888.3264890 doi: 10.1145/3264888.3264890

Sharafaldin., I., Habibi Lashkari., A. & Ghorbani., A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the 4th international conference on information systems security and privacy - icissp,* (p. 108-116). SciTePress. doi: 10.5220/0006639801080116

Sharafaldin, I., Lashkari, A. H., Hakak, S. & Ghorbani, A. A. (2019a). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)* (p. 1-8). doi: 10.1109/CCST.2019.8888419

Sharafaldin, I., Lashkari, A. H., Hakak, S. & Ghorbani, A. A. (2019b). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In *2019 international carnahan conference on security technology (iccst)* (p. 1-8). doi: 10.1109/CCST.2019.8888419

Sharma, M. & Arora, B. (2021). Detection and Prevention of DoS and DDoS in IoT. In P. K. Singh, S. T. Wierzchoń, S. Tanwar, M. Ganzha & J. J. P. C. Rodrigues (Eds.), *Proceedings of Second International Conference on Computing, Communications, and Cyber-Security* (pp. 845–855). Singapore: Springer Singapore.

Shiravi, A., Shiravi, H., Tavallaee, M. & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers and Security*, *31*, 357-374. Retrieved from `http://dx.doi.org/10.1016/j.cose.2011.12.012` doi: 10.1016/j.cose.2011.12.012

Singh, S., Yadav, N. & Chuarasia, P. K. (2020). A review on cyber physical system attacks: Issues and challenges. *Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing, ICCSP 2020*, *2*, 1133-1138. doi: 10.1109/ICCSP48568.2020.9182452

Slowik, J. (2019). Crashoverride: Reassessing the 2016 ukraine electric power event as a protection-focused attack. *Dragos Inc.*. Retrieved from `https://www.dragos.com/wp-content/uploads/CRASHOVERRIDE.pdf`

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M. & Hahn, A. (2015, May). *Guide to industrial control systems (ICS) security - NIST*. Retrieved from `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf`

Tama, B. A., Lee, S. Y. & Lee, S. (2022, 5). A systematic mapping study and empirical comparison of data-driven intrusion detection techniques in industrial control networks. *Archives of Computational Methods in Engineering*. doi: 10.1007/s11831-022-09767-y

Tariq, M. I., Memon, N. A., Ahmed, S., Tayyaba, S., Mushtaq, M. T., Mian, N. A., ... Ashraf, M. W. (2020). A review of Deep Learning Security and privacy defensive techniques. *Mobile Information Systems*, *2020*, 1–18. doi: 10.1155/2020/6535834

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. doi: 10.26599/TST.2020.9010041

Williams, T. J. (1993). The Purdue Enterprise Reference Architecture. *IFAC Proceedings Volumes*, *26*(2, Part 4), 559–564. Retrieved from

https://www.sciencedirect.com/science/article/pii/S1474667017485326 doi: https://doi.org/10.1016/S1474-6670(17)48532-6

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B. & Wessln, A. (2012). *Experimentation in software engineering*. Springer Publishing Company, Incorporated.

Wu, M. & Moon, Y. B. (2017). Taxonomy of cross-domain attacks on cybermanufacturing system. In *Procedia computer science* (Vol. 114, p. 367-374). Elsevier B.V. doi: 10.1016/j.procs.2017.09.050

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. doi: 10.1016/j.micpro.2020.103201

Yang, H., Cheng, L. & Chuah, M. C. (2019). Deep-learning-based network intrusion detection for scada systems. In *2019 ieee conference on communications and network security (cns)* (p. 1-7). doi: 10.1109/CNS.2019.8802785

Yang, T., Hou, Y., Liu, Y., Zhai, F. & Niu, R. (2021). Wpd-resnest: Substation station level network anomaly traffic detection based on deep transfer learning. *CSEE Journal of Power and Energy Systems*, 1-12. doi: 10.17775/CSEEJPES.2020.02850

Yoachimik, O. (2022, 7). *DDoS attack trends for 2022 Q2*. Retrieved from https://blog.cloudflare.com/ddos-attack-trends-for-2022-q2/

Zahid, F., Funchal, G., Melo, V., Kuo, M. M. Y., Leitao, P. & Sinha, R. (2022). 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. (2021). Light-Weight Active Security for Detecting DDoS Attacks in Containerised ICPS. *2021 18th International Conference on Privacy, Security and Trust, PST 2021*. doi: 10.1109/PST52912.2021.9647782

Zeinalpour, A. (2021). *Addressing High False Positive Rates of DDoS Attack Detection Methods* (Unpublished doctoral dissertation).

Zhang, D., Wang, Q.-G., Feng, G., Shi, Y. & Vasilakos, A. V. (2021). A survey on attack detection, estimation and control of industrial cyber–physical systems. *ISA Transactions*, *116*, 1-16. Retrieved from https://www.sciencedirect.com/science/article/pii/S001905782100046X doi: https://doi.org/10.1016/j.isatra.2021.01.036

Zhang, J., Pan, L., Han, Q. L., Chen, C., Wen, S. & Xiang, Y. (2022, 3). Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica*, *9*, 377-391. doi: 10.1109/JAS.2021.1004261

Zhao, J., Shetty, S., Pan, J. W., Kamhoua, C. & Kwiat, K. (2019, 21 Feb). Transfer learning for detecting unknown network attacks. *EURASIP Journal on Information Security*, *2019*(1), 1. Retrieved from https://doi.org/10.1186/s13635-019-0084-4 doi: 10.1186/s13635-019-0084-4

Zhu, B., Joseph, A. & Sastry, S. (2011). A Taxonomy of Cyber Attacks on SCADA

Systems. In *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing* (p. 380-388). doi: 10.1109/iThings/CPSCom.2011.34

Zolanvari, M., Teixeira, M. A., Gupta, L., Khan, K. M. & Jain, R. (2021, Oct). *WUSTL-IIOT-2021 Dataset for IIoT Cybersecurity Research.* Washington University in St. Louis, USA. Retrieved from `https://www.cse.wustl.edu/~jain/iiot2/index.html`