A Process-Informed Approach to Network Intrusion Detection for Industrial Control System

Moojan Pordelkhaki

A thesis submitted in partial fulfilment of the requirements of Birmingham City University for the degree of Doctor of Philosophy

BIRMINGHAM CITY UNIVERSITY

College of Computing

Faculty of Computing, Engineering and Built Environment

Birmingham City University

United Kingdom

January 2025

Abstract

The highly-connected nature of Industrial Control Systems (ICS) has significantly increased the possibility of cybersecurity threats to these systems. Waterfall company's 2023 report showed 218 ICS security incidents, with 25% resulting in tangible consequences, including operational disruptions and equipment damage. This data underscores the criticality of robust ICS security measures. Given that ICS manage essential services, potential compromises could lead to severe disruptions, impacting public health and safety and economic stability.
Network Intrusion Detection System (NIDS) are crucial for securing ICS, providing early threat detection, enhanced network visibility, and invaluable support during incident response. Machine Learning (ML) significantly enhances NIDS capabilities by analysing vast amounts of data to discern normal network behaviour and identify attack patterns. This enables ML-powered NIDS to adapt to evolving threats and identify anomalies with greater accuracy than traditional rule-based systems, all while reducing the occurrence of false positives.
This thesis investigates the potential of integrating both network traffic data and physical process data in the training of ML-based network intrusion detection model. It is hypothesised that this combined approach will yield a more effective detection performance compared to models trained solely on network traffic data. To enable the network intrusion detection model to function solely on network traffic during runtime, the Learning Using Privilege Inform-

ation (LUPI) paradigm is adapted as a key element of the proposed Process Informed Network Intrusion Detection for Industrial Control Systems (PINIDS) framework. The initial phase involves supervised training of a network intrusion detection model using both network traffic and process data. Subsequently, the trained model can be deployed to detect potential intrusions by analysing network data during runtime.

The effectiveness of PINIDS framework for intrusion detection is evaluated using the SWaT dataset, focusing on brute force and unauthorised command message attacks. Various machine learning techniques adopted to the LUPI paradigm are investigated, including Knowledge Transfer (SVM+), Margin Transfer, Transfer Learning, and Distillation. The findings demonstrate enhanced precision and recall balance, leading to improved detection accuracy and reduced false positives and false negatives. Notably, SVM+ achieved a significant 21.47% improvement in F1-score and 49.19% in precision compared to classical ML models, exhibiting consistent performance across experimental runs. While Margin Transfer yielded a modest average improvement in F1-score and precision of 3.3%, it lacked robustness. Distillation proved highly effective, particularly for the DNN model, with a 12.23% F1-score improvement and substantial precision enhancement. Both distilled Deep Neural Network (DNN) and Convolutional Neural Network (CNN) models demonstrated robust performance. Although pre-trained and baseline CNN models performed comparably, the former exhibited a 7.058% F1-score improvement, reduced detection time, and greater stability. These results highlight the potential of transfer learning techniques for enhancing intrusion detection systems.

While Deep Learning algorithms, such as CNN, generally outperform ML algorithms like Support Vector Machines, our findings demonstrate that Machine Learning-based LUPI methods surpass Deep Neural Network-based LUPI

approaches in ICS application with limited training data. The feature-based teaching method employed by SVM+ contributes to its superior performance compared to Deep Neural Network models in this study, effectively leveraging input variable influence for decision-making.

Declaration

I declare that this thesis has been composed in adherence to the guidelines for the Doctor of Philosophy degree. I have composed the thesis independently, and it has not been previously utilised in any previous degree application. The work documented in this thesis has been performed by myself, Moojan Pordelkhaki.

Acknowledgements

I would like to thank my supervisors, Professor Junaid Arshad, Professor Mark Josephs and Dr Shereen Fouad, for their invaluable guidance, support and inspiration. I am truly honoured to have had the chance to be supervised by three incredibly accomplished academics.

I would like to thank Professor Simon Parkinson and Dr Faisal Saeed for their willingness to serve as examiners for this thesis. I am deeply grateful to Professor Wenyan Wu and Dr Ian Rice for their invaluable insights shared during the interviews on the Annual Progress Review meetings. Furthermore, I would like to express my gratitude and recognition to the distinguished scholars, security experts, and individuals who generously provided their knowledge and insights for this research. I am immensely thankful to Birmingham City University for their generous financial support towards my education.

Thank you the exceptional Cybersecurity Research Group for the encouragement you have brought and creating a wonderful atmosphere within the faculty. A handful of profoundly significant friendships have played a crucial role in making these years significantly happier. To all my friends, Thank you so much for the great times and moral support.

My family, including my husband, son, parent, and sister, deserves my deepest appreciation. I am incredibly grateful for your love and support, and I do not know how I would manage without it. Thank you for everything.

Publications

This section presents the research published during the course of this Phd that relates to the content of this thesis, and link it to the relevant chapters.

- Pordelkhaki, M., Fouad, S. and Josephs, M., 2021, November. Intrusion detection for industrial control systems by machine learning using privileged information. In 2021 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 1-6). IEEE.(This publication forms the basics of Chapters 3, 4 and 5)

- Pordelkhaki, M., Arshad, J., Fouad, S. and Josephs, M., 2024, December. A Process-Informed Approach to Network Intrusion Detection for Cyber Physical Systems, under journal review (This submission forms the basis of Chapters 6)

Contents

List of Figures

List of Tables

List of Abbreviations

ARM Association Rule Mining

BAD Behavioural Anomaly Detection

BP Backpropagation

CI Correlation Indices

CIG Correlation Index Generator

CKB Correlation Knowledge Base

CNN Convolutional Neural Network

DBN Deep Belief Networks

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DBSCAN-OD DBSCAN Outlier Detection

DCC Deep Continuous Clustering

DCN Deep Clustering Network

DCS Distributed Control System

DCT Discrete Cosine Transform

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DEPICT Deep Embedded Regularised Clustering

DEN Deep Embedding Network

DL Deep Learning

DNN Deep Neural Network

DNP3 Distributed Network Protocol 3

DoS Denial of Service

DT Decision Tree

DTMC Discrete Time Markov Chain

EM Expectation-Maximisation

ERP Enterprise Resource Planning

FDI False Data Injection

FP False Positive

GAN Generative Adversarial Network

GMM Gaussian Mixture Model

GOOSE Generic Object Oriented Substation Event

HIDS Host-based Intrusion Detection Systems

HMI Human Machine Interface

I/O Input/Output

ICCP Inter-Control Centre Communications Protocol

ICMP Internet Control Message Protocol

ICS Industrial Control Systems

IDS Intrusion Detection System

IED Intelligent Electronic Device

IOA Information Object Address

IOC Indicators of Compromise

IP Internet Protocol

IT Information Technology

KNN K-Nearest Neighbour

LM Levenberg-Marquardt

LR Logistic Regression

LSTM Long Short Term Memory

LUPI Learning Using Privilege Information

MAC Media Access Control

MCA Monitoring-Control Attack

MES Manufacturing Execution Systems

MIS Management Information Systems

ML Machine Learning

MLP Multilayer Perceptron

MODBUS MODBUS protocol

MTD Moving Target Defence

NB Naive Bayes

NID Network Intrusion Detection

NIDS Network Intrusion Detection System

NN Neural Network

OCC One-Class Classification

OCSVM One-Class Support Vector Machine

OT Operational Technology

PCA Principal Component Analysis

PI Privileged Information

PINID Process Informed Network Intrusion Detection

PLC Programmable Logic Control

POMDP Partially Observable Markov Decision Process

PRI-IDS Probability Risk Identification-based Intrusion Detection System

RL Reinforcement Learning

RNN Recurrent Neural Networks

RTI Radio Tomographic Imaging

RTU Remote Terminal Unit

RTU Remote Terminal Unit

SARSA State-Action-Reward-State-Action

SARSA State-Action-Reward-State-Action

SCADA Supervisory Control and Data Acquisition

SD-IDS Stereo Depth Intrusion Detection System

SV Sampled Values

SVD Singular Value Decomposition

SVM Support Vector Machine

SWaT Secure Water Treatment (testbed)

TTPs Tactics, Techniques, and Procedures

TASE.2 Telecontrol Application Service Element 2

TCP Transmission Control Protocol

TD Learning Temporal Difference Learning

UAV Unmanned Aerial Vehicle

UDP User Datagram Protocol

VPN Virtual Private Network

WSN Wireless Sensor Network

Chapter 1

Introduction

ICS are integral to the function of modern society, facilitating critical infrastructure operations that are often taken for granted. These systems underpin essential services such as energy production and distribution, water treatment and distribution, transportation system, and manufacturing processes. The reliable operation of ICS is crucial for maintaining safety and ensuring economic productivity.

ICS are responsible for managing and controlling critical infrastructure operation that are fundamental to daily life. In the energy sector, ICS manage power generation, transmission, and distribution networks. Water treatment and distribution systems rely on ICS for ensuring water quality and safety, while transportation systems, including traffic control, railway signalling, and pipeline operations, depend on ICS for safe and efficient operation. Manufacturing industries use ICS for process control and automation in production lines. These systems maintain process control by regulating parameters such as temperature, pressure, and flow to ensure safe and efficient operations.

The economic impact of ICS is significant, as they enable efficient and optimised operations. By automating tasks and processes, ICS enhance productivity

and reduce the reliance on manual labour, leading to increased output. Furthermore, ICS contribute to reduced cost by optimising resource, minimising waste and improving overall operational efficiency. The impact of ICS on every day life is profound, despite their often-unseen nature. The electricity supply to homes and businesses is reliant on power plants and grids managed by ICS. Access to clean water is ensured by water treatment facilities that depend on ICS for their operations. Transportation and logistics networks rely heavily on ICS for smooth and safe functioning. The manufacturing of essential goods, from food production to automotive manufacturing, relies on ICS-controlled processes.

A disruption to ICS can have cascading effects on society. For instance, a disruption to the ICS managing a city's power grid could result in widespread blackouts, impacting residential, commercial and industrial activities.

Despite their critical role in critical infrastructure, ICS often have inherent vulnerabilities that make them attractive targets for cyberattacks. These vulnerabilities derive from various factors, including legacy systems, increasing connectivity, difficulties in patching and updating, and physical security challenges. Many ICS operate on outdated operating systems with known vulnerabilities that may no longer receive security updates. the increasing connectivity between Information Technology (IT) and ICS networks has expanded the attack surface for ICS. The traditional air gap that once separated these networks is vanishing, providing potential avenues for attackers to traverse between IT and ICS environment. Remote access, while fulfilling maintenance and monitoring purposes, can introduce vulnerabilities if not accompanied with robust security measures.

The operational Constrains of ICS pose significant challenges for patching and updating. Additionally, the geographically distributed nature of ICS, with com-

ponents located in remote areas, makes it difficult to secure every component physically and exposes these systems to the risk of physical tampering, unauthorised access and potential disruption. Once the attackers exploit any of these vulnerabilities and get inside the system, they could manipulate control commands, causing equipment malfunctions, process disruptions or even physical damage.

Two high-profile cyber incidents occurred in 2021. The Colonial Pipeline incident disturbed gasoline supply to the Southeastern United States (Blount 2021). This was a ransomware attack, where criminals used malicious software to block access to Colonial Pipeline's IT in return of the ransom payment. In response to the attack, the company halted operations to ensure the malware did not spread to its Operational Technology (OT). The attacker used a legacy Virtual Private Network (VPN) account which was inactive. Not only did this account still had access to the network, but also it lacked Multi-Factor Authentication control.

An attack on the Florida water supply increased the sodium hydroxide level in the water system before detection (Addeen et al. 2021). Here, attacker attempted to remotely access the plant's control system and change the level of sodium hydroxide level. To do so the adversary used the remote access software deployed on a computer connected to plant's control system and manipulated the control system interface.

The cybersecurity of ICS presents unique challenges due to their inherent characteristics and operational environments. These challenges include dependencies on legacy systems with outdated and often unpatchable operating systems and hardware, leading to exploitable vulnerabilities. Furthermore, the use of specialised, proprietary protocols and systems within ICS hinders the implementation of standard security tools and practices. Real-time operation

constraints necessitate security measures with minimal latency, adding another layer of complexity.

The ICS threat landscape is continuously evolving, with increasingly sophisticated attackers such as nation-states and organised criminal groups targeting these systems to disrupt critical infrastructure or exfiltrate sensitive information. Supply chain vulnerabilities represent another significant risk, as compromised hardware or software can introduce vulnerabilities into ICS environments. Additionally, malicious insiders or unintentional errors by personnel with access to the ICS network pose a significant threat. The convergence of IT and ICS networks expands the attack surface, introducing new entry points for attacks. Traditional IT security tools may be inadequate for monitoring and securing ICS environments, potentially leading to vulnerabilities and blind spots. Whilst eliminating known weaknesses is a critical protective approach, it is also essential to detect undesired activity across the industrial network before any irreversible damage occurs.

As ML techniques enable the automation of anomaly detection and make it more efficient, particularly when handling large datasets, researchers have been examining the use of these methods for intrusion detection, primarily anomaly detection in the constantly changing ICS domain. ML techniques can be divided into supervised, semi-supervised and unsupervised; examples of each of these techniques can be found in the works of Agrawal et al. (2018), K. Yau et al. (2017) and W. Gao et al. (2010). While supervised learning relies on labelled data, a labour-intensive operation that requires domain knowledge, these techniques are known for their high predictability. On the other hand, semi-supervised and unsupervised methods use unlabelled data, however, these techniques present lower predictive performance. The comparative analysis presented by Bernieri et al. (2019) suggested that supervised algorithms used for an-

omaly detection in ICS outperform semi-supervised and unsupervised algorithms. Research into cyber anomaly detection in ICS can be divided into two groups: NIDS and Behavioural Anomaly Detection (BAD). Research on Supervised ML-based NIDS has mainly focused on using industrial network traffic. For example, using network traffic Maglaras and J. Jiang (2014) applied One-Class Support Vector Machine (SVM) algorithm for identifying the attacks, Patel et al. (2017) used a knowledge-based analysis method for detecting and classifying the attacks and Schneider et al. (2018) used an auto-encoder neural network to identify attacks.

However, Suaboot et al. (2020) argued the importance of data analysis on data integration between the collectable data from multiple sources in ICS, and urged for a more holistic perspective toward NIDS for ICS. When an attacker implants a malicious message in a packet payload, altering no protocol structure or communication pattern, ML-based NIDS developed using network traffic may not be successful in identifying this abnormal behaviour. Hadžiosmanović et al. (2014) highlighted the importance of applying knowledge domain and process behaviour in identifying cyber incidents. Studies focused on Supervised ML-based BAD are mostly focused on using process data to learn ML algorithms. Process data in ICS refers to real-time information collected from sensors and actuators within industrial environments. This includes measurements of temperature, flow rate, pressure, valve positions and pump statuses. These data points represent the operational state of physical processes and can provide valuable context for identifying abnormal or malicious behaviour in ICS environments. For example, Junejo and Goh (2016) used process data to train supervised algorithms such as Decision Tree, Naive Bayes and Support vector Machine for detecting attacks and identifying the type of attacks Agrawal et al. (2018) used the rate of change in the process data to train an SVM algorithm

for identifying abnormal behaviour in the system and Hink et al. (2014) deployed different supervised algorithms such as SVM, Naive Bayes and Random Forest for classifying natural and malicious power system disturbance in a smart power grid.
This research combines elements from cyber security and industrial systems with machine learning and aims to enhance NIDS for ICS through the combination of cyber (network) and physical (process) data.This research addresses the primary problem of traditional NIDS systems which rely solely on network traffic because they cannot identify sophisticated or concealed attacks that occur during reconnaissance phases or multi-stage attacks.

1.1 Motivation

ICS interact with the physical environment. Therefore, studying the physical properties of the process under control can provide valuable information about the desired behaviour of the system and its input/output devices. Investigating the process behaviour might be a useful approach to detect process anomalies. However, identifying the source of these undesired behaviours will not be possible without investigating other aspects of the ICS, such as communication between the elements and the process. All abnormal behaviour in the process is not because of cyber-attacks, there are other reasons for a system's unexpected behaviour, such as device or system failure e.g. inaccurate reading of a sensor in a manufacturing plant can cause process deviation from normal behaviour, equally unpredicted environmental changes such as temperature fluctuation and power surges can disrupt normal operation. Differentiating between maintenance problems and cyber incidents will not be achieved through only behaviour-based process analysis. Further valuable information can be derived

from the industrial network communication in ICS. Such information would lead to the design of a NIDS that can uncover cyber-attacks during the reconnaissance period where the attacker is collecting intelligence and has not yet manipulated the physical process.

1.1.1 The Research Gap that Motivates This Research

Supervised ML-based NIDS for ICS have been extensively researched, with a primary focus on leveraging industrial network traffic while often overlooking physical process data. These approaches often rely on network traffic alone, overlooking the contextual insights embedded in physical process data. This limitation enables attackers to execute process manipulation attacks which remain undetected because they do not modify protocol structures. These studies contain two fundamental limitations because they assume uniform attack patterns while disregarding the complex nature of multi-vector attacks. Suaboot et al. (2020) supported the use of data from various sources in ICS and proposed a holistic approach. This is crucial for detecting attacks targeting ICS as an attacker could inject a malicious message within a network packet's payload without violating any protocol or communication pattern, rendering a NIDS unable to detect it. Some researchers emphasised the importance of incorporating domain-knowledge and process context by modelling the physical processes to effectively detect cyber-attacks in ICS (Hadžiosmanović et al. 2014). However, generating an accurate model necessitates a comprehensive understanding of the physical processes and algorithms involved. Other studies have concentrated on supervised ML-based anomaly detection solely utilising process data. The supervised anomaly detection methods achieve acceptable even high results on SWaT datasets when using process data yet they fail to perform well when faced with multiple attacks and reconnaissance activities that requires

focusing on Network traffic. This research addresses these limitations by proposing a hybrid model that leverages process context to enhance NIDS accuracy and reduce false positives. It is important to acknowledge that anomalies can occur due to device or system failures unrelated to cyber-attacks. Therefore, anomaly detection based solely on physical process data cannot differentiate between a malfunctioning sensor and a cyber-attack. Furthermore, NIDS are capable of identifying the reconnaissance phase of a cyber-attack, which typically occurs without impacting physical processes.

This research investigates network traffic and process data integration and their joint interpretation to detect complex cyber intrusions. An ML-based NIDS for an ICS that is developed based on both network traffic and physical process data may outperform the one trained and tested using network data alone. However, because of the different data collection rates from network traffic and process parameters, integrating these two data for the testing period seems to be impractical. First, the frequency of data collection from network traffic is higher (e.g. in milliseconds) than the process data (e.g. in seconds/minutes), which highly depends on the configuration and the nature of the process parameters. Second, network data and process data are being stored in different locations during the run-time. This research aims to address these needs, contributing to research into leveraging the LUPI framework (Vapnik and Vashist 2009b; Vapnik 2006) to detect cyber intrusion in an ICS network as an approach to improve attack detection in the domain. This research intends for designing a process-informed network intrusion detection framework for ICS, and explore the extent to which this framework could aid in reducing the false positives in detecting cyber attacks within industrial networks. The anticipated result of this research is evidence of the role of process behaviour in improving network intrusion detection as an approach to network security in ICS.

1.2 Problem Statement and Research Aims

Even with recent progress—especially methods that use process data to spot unusual behaviour—current research still struggles to detect complex or simultaneous attacks, especially when it relies only on network traffic. The aim of this research project is to explore the potential for improvement in the performance of NIDS to detect cyber attacks by incorporating data from network traffic and process data during training an ML based NIDS.

Although LUPI framework has been used in different domains of research, including computer vision (Momeni et al. 2018; X. Yang et al. 2017; Xu et al. 2015), astronomy (Fouad, Tino et al. 2013; Fouad and Tiňo 2013; Fouad 2013), and medical diagnosis (Shaikh et al. 2020; L. Shen et al. 2020), to our knowledge, the application of LUPI in NIDS for ICS has not been investigated before. Given the successful implementation of the LUPI framework in cybersecurity, as demonstrated by its effectiveness in detecting malicious botnet activities in IT networks (Sapello et al. 2017), and its proven use case in anomaly detection across security applications such as facial recognition, bot detection, and malware detection in IT networks (Celik et al. 2018), the LUPI framework holds great potential in enhancing and complementing the existing NIDS for ICS networks.

In this research, we present an ML-based NIDS that integrates network and process data during the training phase; the model uses network data during the testing phase only. Our method is based on the LUPI framework (Vapnik and Vashist 2009b; Vapnik 2006). LUPI is a supervised ML paradigm that allows one to design a model by integrating additional informative features, known as privileged information, during the training phase.

Our hypothesis is that the proposed approach would enhance the performance of intrusion detection compared to models using only network data for

training. We have evaluated our proposed method of ML-based NIDS for ICS through various ML techniques (including Neural Network) in the Secure Water Treatment (testbed) (SWaT) test-bed, which is an operational scaled-down water treatment plant (Goh, Adepu, Junejo et al. 2016).

1.3 Research Objectives

The main objectives of this research are:

- **Objective 1:** Design an ML-based NIDS for ICS that trains with both process and network data but runs at network level.
- **Objective 2:** Evaluate the practicality and benefit of the LUPI paradigm in ICS using SVM+ (train: network+process; test: network-only).
- **Objective 3:** Compare traditional NIDS with LUPI-based models across single-attack and multi-attack scenarios.
- **Objective 4:** Evaluate robust neural-network for knowledge transfer (e.g. margin transfer, distillation, transfer learning) following LUPI framework, under the same principles, train with both process and network data and run with network data.

1.4 Contributions

There is a large body of work investigating ML-based anomaly detection for ICS. However, to the best of our knowledge, we are the first to bring the implementation of LUPI to ML-based NIDS in ICS domain (Pordelkhaki et al. 2021). This work provides a substantial contribution through the Process Informed Network Intrusion Detection (PINID) framework for NIDS. Proposed

framework is designed with consideration of the cyber and physical characteristics of ICS. PINID suggests the use of network traffic and data processing to train a ML-based NIDS. During the run-time, however, the model would only be provided with network traffic. By following the LUPI paradigm, we have not foreclosed the knowledge that the process data would provide for learning a robust ML-based NIDS for ICS. The proposed framework was evaluated using the machine learning-based privileged information method (SVM+). We have also examined this framework with robust Neural Network approaches, such as Distillation and Transfer Learning. This proposed NIDS can identify multiple types of attacks. This has been showed through rigorous evaluation performed on a subset of attacks that represent all the attack types from the SWaT database. The evaluation results show detection performance and False Positive (FP) rate.

- **Contribution 1:** As identified by (Tsang et al. 2005; D. Yang et al. 2006; Cheung et al. 2007; Valdes et al. 2009; Ponomarev et al. 2015; Yusheng et al. 2017), process-level data provide crucial insights into the behaviour of ICS components, which can be instrumental in identifying anomalies and malicious activities. In this context, our research introduced the PINID framework, a novel approach for NIDS in ICS that considers both cyber and physical aspects. Unlike existing research that primarily relies on network data for intrusion detection, the PINID framework leverages the strengths of both network and process data to enhance the detection of cyberattacks within an ICS. By incorporating process data alongside network data, our framework enabled a more comprehensive and accurate assessment of the ICS security posture. Following the principles of the LUPI framework, we utilised network data for developing a NIDS for ICS, aligning with established practices in the field. However, unlike many

proposed techniques that disregard the valuable information embedded within process data, our approach retained and leveraged this knowledge to train a more robust and reliable ML-based NIDS for ICS.

- **Contribution 2:** To assess the effectiveness of the PINID framework, this study examined a range of attacks that can compromise industrial processes taking into account existing work conducted by MITRE (Alexander et al. 2020) aligned with established hierarchical Purdue Model (Williams 1994) for ICS network. Focusing on brute force IO and unauthorised command message techniques, the evaluation involved attacks targeting both single and multiple field devices (sensors or actuators) within one or more stages of a water purification process. The evaluation utilised the SWaT dataset (Aditya P Mathur et al. 2016a), which comprises ICS experiments conducted in 2016 on a small-scale water treatment plant. This dataset, containing 36 different attack types, provided a comprehensive representation of potential threats to a typical ICS. Among other challenges, our evaluation also highlighted the critical role of labelling in supervised learning environments and developed an approach to address limitations in existing datasets in this respect.

- **Contribution 3:** Considering single attack scenario, we evaluated the performance of PINID framework with common supervised machine learning algorithms (SVM, Logistic Regression (LR), Decision Tree (DT), Multilayer Perceptron (MLP), K-Nearest Neighbour (KNN)) using network traffic data for training and testing. Specifically, the study compared the performance of these algorithms with knowledge transfer technique, SVM+, the SVM algorithm adapted Learning Using Privileged Information (LUPI), where process information supplemented network traffic data. Addition-

ally, the framework's evaluation incorporated robust neural network methods in detection of a single attack. This framework validation included evaluating margin transfer and distillation techniques alongside transfer learning, utilizing a neural network structure. The outcomes of this evaluation demonstrated that knowledge transfer outperforms other ML algorithms in detecting single attack, however, the results from neural network techniques demonstrate that margin transfer technique outperforms all ML techniques in detecting single attack and emerged as the superior technique for single attack detection among other deployed ML techniques. This contribution builds upon the existing body of research including works of Schneider et al. (2018) and Goh, Adepu, Junejo et al. (2016) by using traditional ML methods for intrusion detection from either network or process data but not both in a combined setup.

- **Contribution 4:** Considering the complexity of multi-attack scenarios, which introduce diverse attack characteristics, we evaluated the performance of the PINID framework. This evaluation employed common supervised ML algorithms, including SVM, LR, DT, MLP, KNN, using network traffic data for both training and testing. Specifically, the study compared the performance of these algorithms with a knowledge transfer technique, SVM+, which is the SVM algorithm adapted for LUPI. In this adaptation, process information was appended to network traffic data as auxiliary information. Beyond traditional machine learning methods, the framework assessment also encompassed robust neural network approaches. This involved evaluating margin transfer and distillation techniques, along with transfer learning, all implemented within a neural network structure. The findings indicate that utilising distillation techniques in conjunction with LUPI proves more effective in identifying multiple attack types compared

to other methods. This contribution extends the work of Suaboot et al. (2020) and Hadžiosmanović et al. (2014) proposed holistic ICS security models that could handle various attack modalities but did not assess robustness in scenarios involving multiple attacks.

1.5 Thesis Structure

The remainder of this thesis is organised as follows:

Chapter 2 — Background and related work

Chapter 3 — Materials and methodology

Chapter 4 — A framework for ML-based network intrusion detection in ICS

Chapter 5 — Result and analysis

Chapter 6 — Discussion and finding evaluation

Chapter 7 — Conclusions and future work

Chapter 2

Background and related work

2.1 Industrial Control Systems

The term ”Industrial Control Systems” refers to a collection of several automated control process systems. These systems can control industrial processes through real-time data gathering and monitoring. These include industrial systems like Programmable Logic Control (PLC), Distributed Control System (DCS), and Supervisory Control and Data Acquisition (SCADA) (Fan et al. 2015). ICS are used in various automated industries like the food industry, transportation industry, chemical and petrochemical industry, oil and gas industry, and critical infrastructures such as power plants and water treatment plants (Nankya et al. 2023; Mesbah et al. 2019) and (S. Singh et al. 2020). The significance of safety and security for these infrastructures has increased the value of ICS safety and security. The overall operation of ICS depends on multiple crucial components, each with a significant role (Aykut et al. 2025). Some common components found in ICS are:

- Sensors: These devices gather data from the physical environment. Variables such as temperature, pressure, and flow rate can be measured to

offer real-time data to the control system.

- PLC: PLCs are compact computer devices that gather input from sensors and decide based on pre-established logic.
- SCADA: The SCADA system acts as the central hub for monitoring and controlling industrial processes. The collection of data from diverse devices is displayed for operators and remote control functions are enabled.
- Human Machine Interface (HMI): HMIs refer to graphical user interfaces that visually depict the system's status and enable operators to interact with the ICS. It is common for them to incorporate touchscreens, alarms, and visualisation tools for data.
- Actuators: Commands from the control system are executed by actuators, which perform physical actions. To regulate the industrial process, instructions are received and used to operate motors, valves, switches, and other mechanisms.
- Communication Network: The exchange of data between different ICS components is made possible by these networks. Real-time communication is guaranteed with the option of wired or wireless connections, ensuring seamless system operation.

The architecture of a modern ICS mainly consists of three layers: an enterprise management layer, a supervisory layer, and a field layer. The enterprise management layer mainly includes Management Information Systems (MIS), Enterprise Resource Planning (ERP) systems, Manufacturing Execution Systems (MES) , and other application systems. This layer uses the network communication technology to connect with the Internet, in order to realize the real-time monitoring and management of industrial processes and furthermore as-

sist enterprise-level intelligent decision-making. The supervisory layer consists of process monitoring systems, historical and real-time databases, and a series of operator and engineer stations. This layer is responsible for data acquisition and transmission between the enterprise management layer and the field layer, and controlling field devices based on specific control logics. The field layer includes various types of sensors, actuators, transmitters, and Input/Output (I/O) devices. This layer is mainly responsible for the perception of field information and the manipulation of field devices, and furthermore exchanging digital or analogue data between different field devices through the field bus.

ICS components leveraged ICS-specific protocols for communication over a network, with most of the protocols being point-to-point or broadcast. Currently, numerous protocols are layered on Ethernet, Transmission Control Protocol (TCP), and User Datagram Protocol (UDP), and devices use Internet Protocol (IP)-based networks (Bansal et al. 2024). Feedback control loops are used in control systems to regulate output by detecting and using input from the environment (Åström et al. 2021). The monitoring and evaluation of the physical process helps regulate it based on specific limits set by the user. ICS often integrates extra safety-related logic to back their primary function. PLCs, which directly govern physical processes, are generally created using a basic, logic-oriented methodology. Controllers, with their intricate language, supervise PLCs to control multiple end units and connect with higher-level applications. Collection of such PLCs and supervisory controllers, besides peripheral devices like the HMI, are frequently denoted as SCADA systems (Agha 2024).

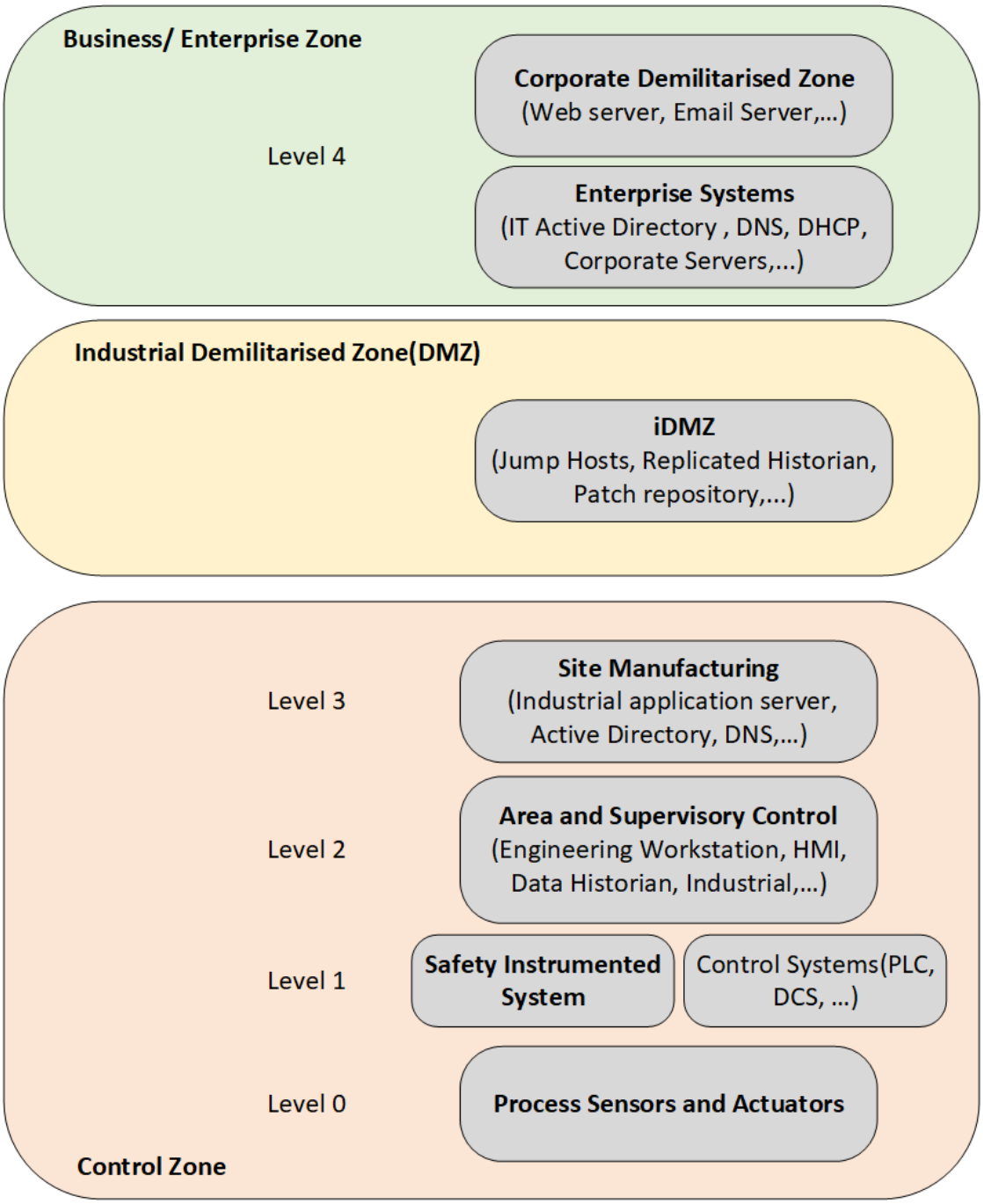

Figure 2.1: ICS Network Architecture

2.1.1 ICS Security

A comparative analysis reveals distinct security requirements for ICS compared to traditional information systems. While the latter emphasises confidentiality, integrity, and availability of sensitive data, ICS security historically prioritised safety, mitigating risks associated with system failures that could impact human life, physical assets, and production continuity(Stouffer et al. 2023). However, the increasing interconnection of ICS with external networks, particularly the internet, necessitates a paradigm shift. Modern ICS must now navigate the complexities of balancing both safety and security imperatives (Ani et al. 2017). Key security prerequisites for ICS in this evolving landscape include:

- The real-time functioning of each physical device can be compromised by even the slightest deviation, leading to industrial disasters.
- The provision of security programs in ICS with multiple sensors and actuators presents a problem because of restricted computing and storage resources.
- ICS must adhere to exact control algorithms to achieve specific production objectives. The likelihood of causing severe accidents increases when these algorithms fail.
- The continuous operation of all ICS equipment presents a formidable obstacle to halting ICS operations for firmware or software updates.
- Connecting ICS to the internet, while offering benefits, significantly increases their vulnerability. Originally designed for isolated networks, ICS protocols face heightened risks in interconnected environments. This exposes critical process data to unauthorised access, leaving organisations

vulnerable to espionage, sabotage, and operational disruptions. The convenience of internet connectivity for ICS comes at the cost of an expanded attack surface and increased cyber threats.

Cyber attacks targeting ICS can be categorized into the standard CIA categories of Confidentiality, Integrity, and Availability. Even though ICS might be prone to comparable CIA attacks as IT systems, the aftermath of these attacks can potentially lead to catastrophic consequences, including jeopardizing personal and social safety and life (Stouffer et al. 2023; Zhou et al. 2020; Slowik 2019). For example in May 2021, critical infrastructure was targeted by cyber attackers when the largest oil pipeline in the U.S. was breached. The threat actors gained access to the corporate network by exploiting an exposed VPN password and proceeded to steal 100 GB of data while also infecting the IT network with ransomware. In order to prevent the ransomware from spreading further, the pipeline was shut down for approximately one week, which impacted airlines due to a shortage of jet fuel and created a sense of social panic regarding a fuel scarcity(Tsvetanov et al. 2021). Also in March 2000, Maroochy Water Services experienced a notorious cyber incident in which an ex-employee hacked into the system and gained control of 150 sewage pumping stations, resulting in the release of one million liters of untreated sewage into local waterways (Slay et al. 2007). Furthermore, due to their utilization of sensors and actuators to interact with the physical environment, ICS are vulnerable to a particular class of cyber-attacks, in addition to the distinctive aftermath caused by a cyber threat. One of the attack types specific to Cyber Physical Systems and ICS is False Data Injection (FDI) attack (Chong et al. 2019).In the event of an FDI attack, adversaries have the potential to exploit sensors and actuators to impose changes in system behaviour that won't affect the observed system state, allowing the manipulation to go unnoticed. Despite the advanced secur-

ity measures in public and enterprise IT networks, industrial networks pose a challenge, requiring bespoke solutions adapted to the control environment. Addressing security concerns in a continual real-time system demands a thorough thorough understanding of network security, control theory, and physical processes. To respond to these challenges, standardisation bodies such as **(NIST, ISA/IEC and ENISA)** have established guidelines and frameworks for securing critical infrastructures. These include general IT security standards, ICS-specific frameworks like IEC 62443, and sector-oriented recommendations tailored to particular industrial domains (Islam 2025; Laan et al. 2025; Maurya et al. 2024).

2.2 Intrusion Detection Systems in ICS

The process of intrusion detection revolves around the examination of events occurring within computer systems or networks. The primary objective is to identify potential incidents that have the potential to violate security policies, acceptable use policies, or standard security practices. (Scarfone et al. 2007) The classification of intrusion detection technologies typically hinges on identified events and the methodology utilised for recognising incidents. IDSs are commonly categorised into network-based known as NIDS and Host-based Intrusion Detection Systems (HIDS) based on their scope and their operation. The fundamental approaches to incident detection encompass detection based on signatures, anomalies, and specifications. Apart from monitoring and analysing events, IDS usually logs information regarding the events, informs the administrator of crucial events through warnings and alarms, and generates suitable reports. The past decade has seen a powerful surge in the research and development of intrusion detection systems for SCADA networks. The rigor-

ous demands of real-time operation and data integrity, combined with consistent traffic patterns and a finite set of telecommunication protocols, mandate the creation and deployment of intricate and dedicated intrusion detection systems. The significant attention given to the security of ICS by both academia and industry can be linked to the emergence of Stuxnet when the importance of intrusion detection technology in protecting the security of ICS has been widely acknowledged (Maglaras and J. Jiang 2014). However, Intrusion detection systems developed for conventional information systems often overlook the unique characteristics of ICS, thereby limiting their efficacy in guaranteeing ICS security. In his study, Mitchell et al. (2014) classified IDS for ICS into different categories based on the detection techniques utilised and the data sources they rely on. With detection techniques, IDS for ICS can be categorised into two distinct groups, namely misuse-based and anomaly-based, focusing on different approaches. Misuse-based IDSs operate by contrasting collected system information against established signatures within a misuse pattern database. This method facilitates the identification and detection of previously documented intrusions. A key strength of misuse-based IDSs lies in their capacity to attain high detection rates for known attack vectors. Anomaly-based IDSs function by contrasting a system's current operational state with its established "normal behaviour pattern." When the discrepancy between these states surpasses a predetermined threshold, an alert is generated. Although anomaly-based IDSs possess the capability to detect a broad spectrum of previously unknown attacks, they frequently encounter challenges related to a heightened rate of false alarms. Besides that, Mitchell et al. (ibid.) have summarised a new subclass of anomaly-based IDS, which is referred to as behaviour specification-based IDS. By utilising industrial control protocols and system behaviour specifications, these IDS can construct the normal behaviour model of a system. On the other

hand, ICS IDS can be classified into two types, network-based and host-based, based on various data sources. Network-based IDS systems rely on network adapters to collect and analyse real-time data of network communication. By utilising various data analysis techniques, these systems can effectively detect and identify global intrusion behaviours. However, it is important to note that this method lacks the capability to precisely pinpoint the specific system or asset that is being targeted. Host-based IDSs concentrate on monitoring documents, processes, and data residing on a designated host to detect any intrusion attempts directed at that specific asset. The taxonomy of conventional intrusion detection systems (IDS) frequently overlooks the specific features of ICS and their intricate linkages with the physical world, as these systems are primarily designed for information systems. By taking into account both the detection techniques and the unique characteristics of ICS, the field of ICS IDS can be further categorised into three specific types: IDS based on protocol analysis, IDS based on traffic mining, and IDS based on control process analysis. The primary focus of the technologies in the first two categories is to detect and analyse standard cyber attacks that are specifically targeted towards ICS. The process of achieving this involves the thorough evaluation of industrial protocols, as well as the careful analysis of traffic data that is generated within industrial control networks. The third category is specifically designed to detect and mitigate semantic attacks that target control systems or physical processes with the intent of causing harm to ICS.

By leveraging protocol analysis, the Intrusion Detection System (IDS) is able to detect and identify malicious attacks within an industrial control network. It achieves this by closely examining the transmission packets and assessing their adherence to the specifications laid out by the industrial protocol. The successful implementation of techniques in this category is highly dependent

on the accurate definition of detection rules. In cases where rule definitions lack precision, a heightened rate of false alarms may occur, potentially leading to undesirable outcomes. Moreover, the process of parsing each transmission packet is known to be time-consuming, adding further delays to the system's efficiency. The IDS take into account traffic mining techniques to address these issues, albeit to a certain extent. The techniques within this particular category aim to create non-linear connections between the patterns of network traffic and the normal or abnormal behaviours exhibited by a system. Despite this, it is important to note that both categories, which are based on traditional information systems, fail to acknowledge the strong link between ICS and the physical world. The existence of this oversight poses a significant threat, as it creates multiple opportunities for attackers to exploit and manipulate industrial process data. Moreover, they can deliberately tamper with the operating rules of field devices, potentially causing severe and irreversible damage to the ICS. The nature of these attacks is in line with protocol specifications, meaning that they do not create any unusual network traffic. As a result, a new type of intrusion detection system has emerged, known as control process analysis-based IDS, which aims to detect and identify semantic attacks like the one described. In the following section, our main objective is to delve into this taxonomy associated with ICS IDS and provide a comprehensive exploration of its components. In the field of ICS, IDS are defined as systems, software applications, or a combination of both that have the role of monitoring the behaviours of ICS. These systems continuously gather and analyse all available data, including sensor readings, network traffic, and system logs, to identify malicious activities or policy violations. The dataset, enriched with domain-specific knowledge encompassing protocol specifications, sensor measurements, host data, system logs and network traffic , facilitates a comprehensive understanding

of industrial control. Immediate notification of a system administrator is crucial upon detection of any malicious activity. Subsequently, implementing remedial actions is essential to protect the ICS from potential damage or destruction. More recently Makris et al. (2025)studies Federated Intrusion Detection Systems (FIDS). These systems perform distributed detection operation across various ICS environments and maintain data privacy. This advocates continuous development of IDS architecture, evolving from their original centralised structure.

2.2.1 Protocol Analysis Based IDS

Within industrial control networks, protocol analysis-based IDSs leverage advanced protocol analysis techniques to detect anomalies. This technology identifies deviations in protocol formats or data packet statuses transmitted within the network. Upon detection of such modifications, the system can subsequently identify any abnormal behaviours exhibited by the ICS.

2.2.1.1 Security Analysis of Common Industrial Protocols

The security analysis of prevalent industrial protocols primarily focused on the reliability and efficiency of ICS during their initial design phase. Due to their relatively closed nature, traditional ICS often neglected the importance of ensuring security for industrial communication protocols. Certain cyber attacks have been observed to target and exploit vulnerabilities in commonly used industrial protocols like MODBUS protocol (MODBUS), Distributed Network Protocol 3 (DNP3), and Inter-Control Centre Communications Protocol (ICCP)/ Telecontrol Application Service Element 2 (TASE.2). MODBUS, introduced in 1979 by Modicon (Modbus 2004), an early and widely adopted bus protocol in industrial sectors, employs a Master/Slave communication model.

In this model, the Master initiates data exchange by transmitting a message to the Slave to request information. However, due to the lack of encryption or authentication mechanisms, MODBUS communication relies on original data, making it susceptible to attacks. Parsing MODBUS addresses and function codes exposes the protocol to potential data theft or manipulation by attackers, introducing significant security vulnerabilities. During the 1990s, the American Electric Power Research Institute (EPRI) introduced the Inter-Control Centre Communications Protocol (ICCP) to facilitate communication between control centres within the power industry. ICCP allows bidirectional communication between clients and servers but introduces security enhancements compared to MODBUS, such as an access control bilateral table. However, security risks persist due to the absence of data encryption and identity authentication mechanisms, making ICCP susceptible to various attacks. Distributed Network Protocol (DNP3), designed to facilitate communication between automation components (Curtis 2005), provides enhanced reliability and incorporates features like data reassembly, fragmentation, verification, priority and link control. However, despite these advantages, vulnerabilities have been identified in the DNP3 protocol. The protocol lacks authorisation or encryption mechanisms, makes it susceptible to potential man-in-the-middle attacks. The preceding analysis underscores that the limited emphasis on security during the design phase has introduced a number of security risks within industrial communication protocols. As a result, a different approach to intrusion detection , grounded in protocol analysis, has been introduced.

2.2.1.2 Public industrial communication protocol analysis based IDS

Apart from the various proprietary protocols, there is also a range of public protocols available in the field of ICS, providing researchers with convenient

access to analyse them. A protocol specification serves to define and outline the accepted communication patterns and the message formats that are related to a specific protocol. Therefore, it is feasible to create intrusion detection mechanisms by taking advantage of these protocol specifications. By implementing this detection system, any behaviour that deviates from the specified protocol can be effectively identified and detected. For instance, Cheung et al. (2007) proposed an intrusion detection technique based on a protocol specification model. This model defines acceptable values for various fields within a data packet and outlines valid relationships between these fields. Moreover, the technique establishes normal communication patterns while considering the specific security requirements, data transmission directions, and transmission ports pertinent to an ICS. Although the method is successful in detecting possible abnormal behaviours, one of its disadvantages is the increased false alarm rate, as it has the tendency of misclassifying emerging normal behaviours as anomalies. Morris, Vaughn et al. (2012) created a customised intrusion detection method specifically designed for Modbus, using the Snort software (Roesch et al. 1999), which is renowned for its capabilities in detecting intrusions. The use of Snort rules allowed for the thorough examination of communication data within industrial networks, resulting in the efficient detection of any unauthorised data. The effectiveness of this approach, however, is highly dependent on accurately defining Snort rules, as they directly impact the accuracy of detection. Researchers, in their quest for agile development, have dedicated their efforts to refining and enhancing traditional IDS to make them more compatible with ICS. Bro (Paxson 1999), which was developed by the University of Berkeley, is an IDS that primarily focuses on capturing network packets through bypass monitoring. By extracting relevant events from the packets, the system can employ a protocol parser that can effectively analyse protocols across various network layers. As a

result, the system could to identify any potential intrusions by analysing these events using policy scripts. H. Lin, Slagell, Di Martino et al. (2013), specifically, contributed to the progress of Bro by implementing a packet parser that was specifically designed to handle industrial protocols, including DNP3. The main function of this enhanced parser is to carefully examine the valid values of different fields within a packet, thereby allowing the formulation of security policies that are coherent with the protocol. In addition, the system has the capability to parse multiple protocols that are used in ICS, which expands its applicability beyond just DNP3.

2.2.1.3 Proprietary Industrial Communication Protocol Analysis Based IDS

The development of IDS techniques in specific industries involves the use of certain proprietary industrial protocols, besides public protocols. Hong et al. (2014) performed a thorough analysis of automatic systems within smart grid substations. The purpose of the analysis was to detect any anomalies or malicious behaviours present in the multi-cast messages. To achieve this, the IEC 61850 standards, which encompass the Generic Object Oriented Substation Event (GOOSE) and Sampled Values (SV) were utilised. Leveraging proprietary protocol specifications, this method has demonstrated efficacy in identifying and detecting various malicious attacks, including Replay Attacks, Packet Tampering, and Denial of Service (DoS) attacks. The preceding analysis suggests that protocol analysis-based IDS systems predominantly employ misuse-based intrusion detection techniques. However, the comprehensive analysis of all packet contents during detection can hinder IDS efficiency. Consequently, researchers propose combining misuse-based and anomaly-based techniques to enhance intrusion detection mechanisms for ICS. Y. Yang et al. (2013) proposed a method that initially employs misuse-based detection technology to

compare observed system behaviour against intrusion patterns within a database, enabling rapid identification of known attacks. Subsequently, anomaly-based technology is utilized to scrutinize the remaining data for the detection of unknown attacks. The experimental results have proven that these approaches are highly effective in improving the accuracy and efficiency of ICS IDS detection. In addition, the effectiveness of intrusion detection can be greatly enhanced by combining protocol analysis-based IDS with traffic analysis. The improvement in intrusion detection accuracy can be achieved by transferring the rules extracted from communication patterns outlined in the specifications of protocols in ICS network traffic and specific business logics to the traffic analysis module. Hadeli et al. (2009) presented an intrusion detection scheme for power systems based on this approach. By extracting network traffic patterns from predefined protocol specifications and formal system descriptions, the scheme generates comprehensive traffic models encompassing both legitimate and illegitimate activities. For instance, the model flags anomalies in two scenarios: when an Intelligent Electronic Device (IED) fails to transmit a GOOSE control message is sent to a multicast address that doesn't correspond with the device's Media Access Control (MAC) address. These extracted traffic rules are then inputted into Snort and converted into Snort rules, enabling the detection and reporting of any predicted but unobserved traffic. In their work, Yusheng et al. (2017) introduced an innovative algorithm called Stereo Depth Intrusion Detection System (SD-IDS) , designed for real-time deep inspection of Modbus TCP traffic. The SD-IDS algorithm is composed of two primary components, which are rule extraction and deep inspection. The rule extraction module is primarily responsible for identifying and extracting semantic relationships between key fields within the Modbus TCP protocol. Conversely, the deep inspection module leverages these extracted relationships, in conjunction

with real-time traffic data, to perform effective anomaly or intrusion detection.

2.2.2 Traffic Mining Based IDS

Many intrusion detection systems (IDS) that rely on protocol analysis encounter significant drawbacks, such as their limited ability to identify unknown attacks and the extended time to analyse data packets. Researchers have been focusing on the development of traffic mining-based intrusion detection techniques to address these issues, albeit to some extent. One of the notable advantages of this approach is the enhanced efficacy in identifying a broad range of unknown attacks. One of the defining characteristics of ICS is that they have fixed operational objectives, meaning that their goals and objectives remain constant and do not change. The stability of the ICS traffic is maintained by these characteristics, resulting in a relatively stable flow under normal circumstances. The significance of traffic data cannot be emphasised enough, as it serves as vital information that accurately reflects the security status of ICS. This opens up new opportunities for the development and implementation of intrusion detection technology based on traffic mining techniques. Traffic mining-based IDS systems gather data from various segments within ICS networks and employ data mining techniques, including neural networks, decision trees, Bayesian classifiers and other algorithms, to analyse the collected data. Alternatively, they may utilise data analytic methods, such as statistical analysis, on the gathered data. The primary objective is to identify and flag any anomalous behaviours within industrial networks. In the study conducted by Yusheng et al. (2017) , a novel approach for intrusion detection was introduced. This approach, which is focused on traffic mining, involves the extraction of several data elements including the source and destination IP address, transport-layer protocol, source and destination port, and the average time interval between adjacent packets. Us-

ing data mining techniques then allow distinction between normal and abnormal behaviours which supports the identification of various types of intrusions such as Replay and Packet Tampering. In their study, Hou et al. (2012) introduced a novel approach that focused on the utilisation of probabilistic Principal Component Analysis (PCA) to identify abnormal network traffic patterns. They identified that one of the main reasons for false alarms is the identification of random burst traffic. Through the construction of a probabilistic Principal Component Analysis (PCA) model for the traffic matrix, Hou et al. (ibid.) have extensively analysed the influence of random burst traffic on PCA. An Iterative Variational Bayesian algorithm was employed to estimate model parameters, which were subsequently used to determine the distribution function of the traffic matrix rank. Observed rank changes served as the primary metric for detecting abnormal traffic within ICS. Experimental results demonstrated the effectiveness of this method in mitigating the impact of random burst traffic on intrusion detection. Artificial Neural Network (NN) are a powerful data mining technique that simulates the cognitive processes of the human brain, making them an effective tool for extracting valuable insights from data. In the context of ICS, this technique is utilised to thoroughly analyse extensive datasets, resulting in the successful detection of intrusions that were previously unknown. Neural Networks play a crucial role in traffic mining as they establish non-linear mapping relationships between various traffic features and the security states of the system, whether it is normal or abnormal. This mapping is achieved through rigorous model training. As a result, the data is classified by these trained models, allowing for the effective identification of abnormal traffic and malicious intrusions in ICS. Vollmer et al. (2009) conducted their work by training a neural network model. During the training process, they extracted various network traffic features, including packet size, Internet Control Mes-

sage Protocol (ICMP) protocol ID, ICMP sequence number, ICMP code, ICMP type, IP protocol ID, IP protocol option, and IP survival time. These features were then used to construct input vectors for the model. Once the feature normalisation process was completed, the error Backpropagation (BP) algorithm was employed to train the neural network model. By extracting real-time network traffic features during the detection phase, input vectors were formed. These input vectors were then classified using the NN model, providing the capability to detect attacks such as Denial of Service (DoS) and eavesdropping. In a subsequent study, Linda, Vollmer et al. (2009) presented a novel feature vector extraction technique based on a sliding window approach, enabling dynamic and precise extraction of network features. The analysed features encompassed the number of IP addresses within a packet sequence, the packet counts associated with a single IP address, the average time interval between consecutive data packets, window duration, data transmission speed, the count of observed protocols within the window, and the total number of identification codes. Besides this, the researchers employed a comprehensive method that effectively combined the techniques of BP and Levenberg-Marquardt (LM) in order to detect abnormal traffic, resulting in a satisfactory level of intrusion detection accuracy.

Ashfaq et al. (2017) introduced a highly effective semi-supervised learning mechanism for Neural Networks (NN) that reduces the reliance on labelled data. The initial step in the training process of this method involves establishing a fuzzy classifier, which is implemented as a neural network model with randomly initialised weights. A small set of labelled data is used to train the fuzzy classifier. Subsequently, the fuzzy classifier is employed to classify the unlabelled data, yielding a membership vector as the final output. Each entry in the input vector represents the degree of belonging to a specific category. As a part of the

model training phase, data that contains varying degrees of ambiguity is incorporated into the original training set. The fuzzy classifier is then retrained using this mixed data.

The process of training a neural network model is time-consuming and requires significant computing resources. In order to address this limitation, researchers proposed a fuzzy logic based approach (Linda, Manic, Vollmer and Wright 2011; Linda, Manic, Alves-Foss et al. 2011; Linda, Manic and Vollmer 2012). The researchers in the study by Linda, Manic, Vollmer and Wright (2011) opted to employ fuzzy rules for modelling the normal behaviour patterns exhibited by ICS. By utilising an adjusted online nearest neighbour clustering algorithm, it is possible to extract the fuzzy rules from the network packet sequence. Embedded sensors can easily handle the computational demands of this learning method, as it requires minimal computing resources. While performing the detection process, the scheme utilises the outputs of multiple fuzzy rules to compute the degree to which the input vectors align with normal behaviour patterns, ultimately enabling the identification of intrusions. In order to enhance the accuracy of intrusion detection, the researchers integrated TYPE-2 fuzzy logic into the model in their follow-up work(Linda, Manic, Alves-Foss et al. 2011). The integration aimed to mitigate the detrimental impact of uncertainties on overall system performance while enhancing the accuracy of network security status monitoring by sensors. Linda, Manic, Alves-Foss et al. (ibid.) developed a TYPE-2 fuzzy logic-based IDS to incorporate domain knowledge into specific industrial environments and network systems. The IDS aimed to establish correlations between intrusion likelihood and network communication characteristics. Experimental results indicated that the architecture enables adaptive algorithm threshold adjustments to enhance the intrusion detection accuracy.

It is important to note that SVM was utilised by several researchers in intrusion

detection for ICS. Through the application of kernel functions, linearly inseparable traffic data could be mapped into a higher-dimensional feature space. Following that, they built a super-plane which was highly effective at distinguishing between behaviours that were normal and those that were abnormal. In their study, Maglaras and J. Jiang (2014) introduced an intrusion detection algorithm for ICS that utilises the One-Class Support Vector Machine (OCSVM) technique. The exceptional aspect of this algorithm lies in its ability to be trained offline, with no labelled training data or any prior knowledge of attack categories, which makes it truly unique. This method can create traffic models for multiple protocols, enabling the detection of various intrusion behaviours targeting ICS systems. Some examples of such behaviours encompass Man-in-the-Middle attacks as well as SYN Flood attacks. While the traditional SVM approach can effectively classify normal and abnormal behaviours, it falls short in accurately classifying the specific anomalies. Consequently, Luo (2013) devised a method for intrusion detection that relies on multi-class SVM, in which several SVM classifiers are integrated to accurately identify the category of an intrusion. In their study, Javaid et al. (2016) introduced a novel deep learning technique for effectively differentiating between normal and abnormal traffic data. The initial stage of this process involves the utilisation of a sparse auto-encoder, which is a method employed for unsupervised learning. The unsupervised learning network is composed of three layers, namely the input layer, the implicit feature layer, and the output layer. The output layer's ability to accurately reproduce the input data can be improved by making adjustments to the network parameters. Subsequently, the training features and the labelled data are employed in the training process to train the classifier, ultimately leading to the successful completion of classification tasks.

A unique algorithm was proposed by Aghdam et al. (2016) which selected net-

work traffic features for intrusion detection using colony optimisation which mimics the behaviour of ant colonies. By removing unnecessary or invalid features, this algorithm significantly improves both the efficiency and accuracy of intrusion detection. This algorithm distinguishes itself from traditional IDS methods by detecting a larger number of attacks and minimising the computing workload. Tsang et al. (2005) developed a carefully planned multi-agent architecture to address the intricate challenge of detecting and defending against intrusions in extensive switching networks. The utilisation of the improved ant colony clustering model, a highly effective biological heuristic learning model, by the authors in this architecture improved the overall efficiency of the system. Using a heuristic search technique in this model helps to generate clusters that are approximately optimal using nearest neighbour, with a specific focus solely on the normal data. The primary goal of this procedure is to identify and extract clusters, which can be further converted into fuzzy rules. Each cluster corresponds to a fuzzy rule in a one-to-one manner. Then, the test data is processed through a series of fuzzy rules to gauge its similarity to regular behaviour. Kiss, Béla Genge et al. (2014) organised and used the data gathered from ICS in a timely manner in their study , where they suggested a unique intrusion detection method that employs clustering algorithms to efficiently detect possible attacks on ICS. The researchers, Caselli, Zambon, Petit et al. (2015) and Caselli, Zambon and Kargl (2015), uncovered a crucial finding about network traffic attacks in their study. More specifically, the researchers found a new form of attack called sequence attacks that had not been known until now. These attacks primarily revolve around the sending of incorrect messages through industrial communication systems. The primary objective is to disrupt the field devices intentionally, causing malfunctions and potentially impacting the physical processes they regulate. The recommended strategy for identify-

ing sequence attacks involves converting network traffic traces into event lists that are organised chronologically. Next, the expected patterns of communication sequences are represented using a model called the Discrete Time Markov Chain (DTMC). The last stage comprised analysing traffic data using the DTMC model, which aided in detecting sequence attacks. As stated by Ferling et al. (2018), the author believed that managing sequence-aware intrusion detection models can be a complex task because of their large size, leading to the necessity of time-consuming traffic analysis. As a result, they suggested a method that includes building smaller traffic models. These models are created by merging multiple states in the DTMC model. The only distinction between the merged states is the range of Information Object Address (IOA)s utilised in the IEC-104 protocol. The use of smaller models is a key factor in achieving effectiveness in reducing complexity and maintaining detection accuracy for most sequence attacks. The vulnerability of the Modbus TCP protocol to cyber attacks, as explained by Marsden et al. (2018), is primarily because of the lack of encryption and authentication. To address this issue, the authors proposed a potential solution known as the Probability Risk Identification-based Intrusion Detection System (PRI-IDS). This system is specifically designed to detect and identify replay attacks by effectively analysing Modbus TCP/IP network traffic. The method involves assigning predefined risk values to traffic data as the initial step. Afterward, certain time intervals of the data are stored in cache, and risk values are calculated for those stored intervals. The system detects potential replay attacks by identifying cached periods with risk values that differ by over 1 standard deviation from the average value. The study conducted by Dong et al. (2018) introduces an intrusion detection approach for industrial networks, which relies on traffic feature maps. The extraction of salient traffic characteristics leverages information entropy to generate traffic feature vectors.

Subsequently, a multiple correlation analysis is performed on these vectors to establish a comprehensive relationship map between the features. The Discrete Cosine Transform (DCT) and Singular Value Decomposition (SVD) approaches are then applied to create a database of perceptual hash digests for both normal and abnormal traffic feature maps. Finally, intrusion detection rules are extracted from this database. This method effectively models the periodic patterns observed in industrial network traffic, transforming textual traffic data into valuable numerical information and providing innovative solutions for ICS IDSs.

2.2.3 Process analysis based IDS

Process analysis-based IDSs leverage the information and unique characteristics of ICS for intrusion detection, distinguishing them from IDSs designed for conventional IT systems. This category encompasses techniques such as control command analysis-based, process data analysis-based, and ICS physical model-based IDS.

2.2.3.1 Command Analysis Based IDS

Within ICS, control commands constitute a critical element, along with other essential components. In order to achieve their attack objectives, adversaries have manipulated the control commands. One effective method for detecting intrusion behaviours in ICS is to analyse control commands. Carcano, Fovino et al. (2010) presented an innovative technology for IDS in their research. Their approach involved creating a brand new language that was specifically designed to describe and communicate accurately the power grids' commands. This study outlined the detection features employed by this innovative IDS

technology. Subsequently, the approach utilized two distinct techniques for Modbus packet analysis. The initial method involved using a single packet signature-based approach, wherein the identification of unauthorised packets sent by PLCs or Remote Terminal Unit (RTU)s was accomplished by analysing the meaning of control commands. An alternative approach, known as the state-based strategy, prioritised monitoring the states of ICS in order to detect intrusions, as the system could be pushed into critical states by invalid control commands. Similarly, H. Lin, Slagell, Kalbarczyk et al. (2013) introduced a methodology focused on distributed ICS for conducting semantic analysis on control commands. By leveraging prior knowledge of network and physical infrastructure in power grids, this technique can expect the repercussions of control commands and thus unveil the intentions of potential attackers. Within the semantic analysis framework, one crucial component is the analysis of network packets from ICS using Bro, which allows for the extraction of control commands. The initial observation from the proposed approach was that an attacker can circumvent the conventional IDS and exert control over the system's critical state by merely opening three outgoing lines. Furthermore, the study provided empirical support for the time saving and reliable intrusion detection advantages obtained by conducting semantic analysis on control commands.

2.2.3.2 Process Data Analysis Based IDS

Intrusion detection systems (IDS) employing control process analysis leverage the semantic information and unique characteristics of ICS for effective intrusion detection, differentiating them from IDS designed for traditional IT systems.

W. Gao et al. (2010) conducted research and categorised three types of attacks that can be directed towards ICS: response injection, Denial of Service (DoS),

and command injection. These researchers suggested a method for monitoring behaviour that uses a model of an artificial neural network and includes information about the physical properties of the system being controlled. The primary purpose of adopting this method is to ensure the accurate identification and effective mitigation of any potential response injection attacks. The proposed approach clearly indicated that artificial neural network is a powerful mechanism for detecting response injection attacks.

The study conducted by Carcano, Coletta et al. (2011) involved the utilisation of different process variables to characterise the system state. To achieve their objective , they have utilised a formal modelling language for their proposed state based intrusion detection. Their novel intrusion detection technique, predicated based on the evaluation of the differences between the current system state and critical system states. While the modelling language primarily used for the Modbus protocol, it could be readily adapted to encompass other industrial protocols. The system provides a virtual representation of the physical system for the IDS, enabling comprehensive monitoring of both environments. Moreover, the language encompasses definitions of critical states, multiple danger levels for ICS, and methodologies for quantifying the disparities between system states. During the detection process, the proximity between the current state and critical states is computed to ascertain their relationship. If the proximity exceeds a predetermined threshold, an alert is generated.

Through the analysis of the semantic elements of process variables, Hadžiosmanović et al. (2014) discovered a method to identify intrusions in ICS.To accomplish this task, they follow a three-step method, starting with the extraction of the value of the current process variable from network traffic. Using these features they designed a model that successfully detects control process-related intrusions. The authors showed their plans to strengthen their intrusion detection

efforts by gathering more specific contextual information. One way to achieve this is by delving into a broader array of structural protocols and system configuration files.

In the study conducted by Krotofil et al. (2015), some argued strict adherence to specific physical laws is required in order to maintain the values of industrial process variables. The authors proposed an innovative method by running a light attack mechanism on field device microcontrollers, enabling the manipulation of process data. The attack mechanism extracted the noise characteristics from the process value patterns of the field devices. Based on the captured noise characteristics and the identified pattern, the attack mechanism can produce a series of values that are deceptive yet appear credible, serving as a substitute for the genuine process variable values. The authors have presented a cluster entropy-based detection method. This method can effectively confirm the logical flow and progression of the value patterns linked to a relevant process variable and identifies intrusion behaviour.

The research conducted by Kiss, Bela Genge et al. (2015) suggests employing a Gaussian Mixture Model (GMM) to identify attacks targeting data transmitted to PLCs. This scheme utilises the GMM to perform data clustering. The Expectation-Maximisation (EM) algorithm is at the heart of the training process, shaping its methodology. The most accurate classification for each measurement is obtained only through a thorough examination and extensive analysis. Observations that deviate from the typical clusters are identified as outliers and evaluated accordingly. Abnormal clusters typically exhibit much lower data densities compared to normal clusters. A soft-classification model like the GMM can provide the confidence level of the association between each measurement and a specific cluster. It can be deduced from the experimental results that the GMM is more successful at identifying intrusions on ICS com-

pared to traditional k-means clustering algorithms. Colbert et al. (2016) introduced a method for detecting intrusions in ICS that specifically targets the control process. In order to enhance the capabilities of the traditional ICS IDS, two detection methods were proposed that specifically target control processes. The main difference between this mechanism and traditional anomaly-based IDS lies in its emphasis on the key role of the man in the loop to define the process variables in an ICS. A positive aspect of having the operator is their specialised knowledge of the nuances of ICS. The regular monitoring of critical process variables guarantees the timely detection of any deviations beyond their thresholds. When this event happens, an alarm sends a notification, promptly alerting the relevant individuals to take the necessary actions. The authors proposed a technique for identifying intrusions by analysing the process network parameters, which are determined through collaboration between the network engineer and the ICS operator. The existence of these parameters may imply the absence of critical control elements or an exceptionally large amount of traffic, both of which are not commonly found in typical industrial settings. Although the critical process variables may indicate significant issues, it is crucial to acknowledge that the process network parameters can still trigger alerts for potential malicious system behaviours.

In their study, Moya et al. (2018) shed light on the emergence of Monitoring-Control Attack (MCA) , a type of attack that is highly dangerous and poses a significant risk to ICS. Fabricating sensor measurements plays a crucial role in these attacks as they are utilised to manipulate control signals in a feedback loop. MCAs are highly likely to occur because of their low cost and their ability to cause considerable damage to ICS. It's difficult to identify MCAs since they often hide within normal sensor data. In this article, an extensive study was conducted to devise a semantic analysis framework for intrusion detection sys-

tems (IDS) in power grids. The main objective of the framework was to address efficiently the risks associated with Monitoring-Control Attacks (MCAs). The framework is made up of two modules, the Correlation Index Generator (CIG) and the Correlation Knowledge Base (CKB), which work together at the same time. The first one is mainly used for indexing related MCAs, while the second one is regularly updated based on changes in attacks' Correlation Indices (CI), serving different functions. The framework has advanced features that enable it to accurately identify and detect malicious cyber attacks (MCAs) and provide immediate estimates of the possible impact of these attacks.

2.2.3.3 Physical Model Based IDS

A physical model can effectively represent the evolution of an industrial control system. Integrating a physical model with prediction mechanisms enables accurate forecasting of expected system outputs (Patton 1995). Comparing observed system outputs with expected values generates a residual series. Statistical analysis of this residual series facilitates intrusion detection. Under normal operating conditions, system residuals typically approach zero. However, during an attack, observed outputs deviate significantly from initial expectations. The behaviour of ICS was represented by Cárdenas et al. (2011) using a state-space model that incorporates approximate linearity. The model suggests that a system's current state is a function of its previous states and control inputs. Utilizing the constructed state-space model enables real-time prediction of sensor measurements. Subsequently, observed sensor data is compared against these predictions, and the resulting residuals are used to detect malicious attacks targeting the ICS. The authors proposed two detection methods: sequence-based and change-based detection. Sequence-based detection, drawing from optimal stopping theory in sequence analysis, aims to rapidly identify

anomalies by determining the minimum sequence required for detection. Conversely, change-based detection focuses on identifying potential changes with uncertain timing by comparing residuals or accumulated residuals against a predefined threshold to detect shifts from normal to anomalous states. Sridhar et al. (2014) developed a model that aimed to enhance the security of smart grids. This model integrated knowledge from power systems to create an intrusion detection and mitigation mechanism. By utilising this mechanism, the model was able to identify various attacks, including data injection, by forecasting the generation load. Y. Liu et al. (2011) identified a novel data injection attack targeting state estimation in power networks. This attack involves the systematic injection of false data into the system, aiming to induce a system crash. However, the attacker closely monitored the volume of the injected data , maintaining it below the threshold and enabling them to evade the stateless intrusion detection mechanism. This marks the initial stealthy attack against ICS. Since then, there has been a rise in stealthy attacks across a range of industrial control scenarios, for instance, chemical process control (Cárdenas et al. 2011) and industrial wastewater treatment (Amin et al. 2012).

The study by Urbina et al. (2016) concluded that current intrusion detection technology lacks effective detection capability for stealthy attacks. This article focuses on the authors' research into reducing the consequences of stealthy attacks. Although it may be difficult to detect these attacks, their effects can be partially minimised by correctly configuring different detection methods and metrics. Theoretical analysis and experimental validation have confirmed that the detrimental effects of stealthy attacks can be mitigated by employing Urbina et al. (ibid.) newly introduced metrics: "the expected time interval between false alarms and the maximum deviation a stealthy attack can cause". These metrics quantify the impact of stealthy attacks. Further research on stealthy

attacks has focused on their execution within specific ICS systems (Kleinmann et al. 2018) or the investigation of their consequences on more complex systems (Kung et al. 2016). However, the urgency to identify stealthy attacks on ICS continues to be a pressing concern for further investigation. In their study, Tian et al. (2018) examined a more advanced scenario of false data injection (FDI) attacks, specifically concentrating on the estimation of the state model in smart grids. The researchers focused on determining if the adversary could identify the implementation of Moving Target Defence (MTD) against FDI before initiating their attack. Their research presents and categorises this specific variant of FDI attacks. They developed a hidden MTD approach that successfully makes itself undetectable to potential attackers, enhancing the stealthiness of MTD even more. By employing a hidden MTD, adversaries can be tricked into initiating ineffective attacks, ultimately increasing the probability of their being discovered. It was determined that the concealed MTD can manage the power distribution of the entire grid, just like the conventional MTD. Myers et al. (2017) highlighted that ICS typically enforce strict task execution order and quantity, resulting in unique task flows for each control system. Consequently, they proposed an ICS attack detection method based on process mining. This method extracts a control process model for the ICS by monitoring and analysing control device log files. Subsequently, it employs consistency detection to identify abnormal system behaviour that deviates from the constructed process model. Furthermore,Samara et al. (2024) pointed out that detecting malware in IIoT-based ICS is becoming more and more dependant on deep learning, and reinforced the need to integrate modern AI techniques into IDS research.

2.3 ML Application in IDS in IDS

Machine learning techniques are widely employed in the development of cybersecurity solutions for ICS. The main goal of these methods is to identify network intrusions by analysing packet information from network traffic or detecting anomalies in physical processes using data that represents the system's physical behaviour. The classification of machine learning and deep learning techniques can be divided into four major categories: Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, and Reinforcement Learning. The majority of the current literature on intrusion detection focuses on the initial two domains. The fundamental distinction between these approaches lies in their utilisation of labelled training data. Unsupervised approaches exclusively rely on the normal behaviour exhibited by the ICS, thereby eliminating the need for labelled data.The implementation of this approach requires training clustering algorithms (e.g. k-means, DBSCAN) with normal process behaviour data e.g. stable Modbus traffic patterns and expected PLC command sequences. Any anomaly that deviate from these cluster identify potential intrusions such as false data injection and unauthorized PLC reprogramming. In supervised methods, the training data contain both normal and abnormal (attack) behaviours. By employing semi-supervised techniques, one can make use of both labelled and unlabelled data, taking into account the fact that labelled training data is restricted, whereas unlabelled data is abundant and readily accessible. The dynamic nature of cyber threats and the continuous advancements in machine learning techniques employed to enhance the security of ICS are worth mentioning. Additionally, the ongoing efforts in research and development are crucial in addressing the difficulties arising from the limited availability of labelled training data and in discovering new methodologies that can easily adjust to the dynamic nature of cyber threats in industrial settings.

Every one of the approaches mentioned earlier has its own merits and limitations. The use of unsupervised learning enables the detection of zero-day attacks without the requirement for labelled training data, thereby eliminating the reliance on attack data. One notable disadvantage is its inclination to generate a high number of false alarms ((Nader et al. 2014a; Nader et al. 2014b)). Nevertheless, supervised learning algorithms offer greater reliability in detecting attacks, albeit necessitating labelled data encompassing both normal and attack instances. Regardless of this requirement, a supervised approach can accurately detect additional instances of attacks with only a few examples. The study conducted by Junejo and D. Yau (2016) demonstrated that supervised algorithms outperformed other classifiers, exhibiting high precision and recall rates while minimising false positives. Nevertheless, it is crucial to acknowledge that these methods may not effectively identify zero-day attacks. Within the field of IDS for ICS, there is a subset of techniques that has received as little focus, including one-shot learning (D. Wu et al. 2012), (Krishnan et al. 2015) and zero-shot learning (Romera-Paredes et al. 2015), (Socher et al. 2013). When discussing machine learning, the concept of "one-shot learning" describes a scenario where the labelled training data includes just one instance of each attack type. Zero-shot learning is particularly difficult due to the lack of labelled training data that contains specific attack instances. Attacks that fall under the category of zero-day attacks are those that are not available training data. Hence, the effectiveness of this learning approach depends on its capacity to detect zero-day attacks and leverage knowledge obtained from previous attacks. This approach offers a practical technique for ICS that has the potential to yield fewer incorrect detections when compared to unsupervised methods. Leveraging data from known attacks that can be safely simulated within a controlled ICS environment it has the ability to identify zero-day attacks. Hence, zero-

shot learning emerges as a promising strategy for intrusion detection systems (IDS) in ICS, effectively fusing supervised and unsupervised methodologies to accomplish favourable results.

2.3.1 Supervised ML Techniques

Supervised Learning employs labelled training data to construct predictive models. Each training instance consists of a feature vector, denoted as x, and a corresponding class label, y. The goal of Supervised Learning algorithms is to learn a function, f, that accurately maps input features to their respective classes. Once trained, the model can predict labels for unseen test data. Supervised Learning techniques are broadly categorized into classification and regression. Classification tasks involve discrete class variables, while regression deals with continuous ones. IDS are often modelled as classification problems, where the class variable represents the presence or absence of attacks. More specifically, The supervised IDS methods in ICS environments use classification techniques to identify normal network traffic or sensor data versus specific attack types such as replay and command injection. One-Class Classification (OCC) pertains to scenarios with only one class label. Traditional statistical approaches, utilizing metrics like mean and standard deviation, have been employed for behaviour-based IDS (Kwon et al. 2015) and (D. Yang et al. 2006). However, these methods suffer from limitations in automation and scalability due to their parametric nature. Defining statistical tests for complex systems with numerous interdependent sensors and actuators is challenging and may result in unacceptable false positives. In contrast, Machine Learning (ML) and Deep Learning (DL) offer non-parametric alternatives that exhibit greater automation potential and technique diversity. ML approaches can be grouped into

discriminative/conditional, generative, and tree-based methods, each with its own strengths and weaknesses.

2.3.1.1 Discriminative/Conditional Approaches

- **Support Vector Machines (SVMs):** These linear, non-probabilistic binary classifiers project data points onto a higher-dimensional feature space and construct a hyperplane to separate the two classes. SVMs are known for their robustness in classification tasks, including IDS (Ahmad et al. 2014). In ICS,, SVM have been used to detect malicious manipulation of sensor data and abnormal network traffic pattern, for example Agrawal et al. (2018) trained SVM on process variable rates of change to identify abnormal behaviour in industrial processes, while Hink et al. (2014) used SVM to classify disturbances in smart power grids.

- **Neural Networks (NNs):** NNs excel in estimating functions with numerous input variables. They consist of input, output, and hidden layers, trained to learn non-linear decision boundaries for class separation. NNs have found applications in IDS ((Al-Jarrah et al. 2015)). In ICS, NNs are particularly effective at capturing the non-linear relationship between sensor readings and control states. Vollmer et al. (2009) showed that NNs trained on Modbus network traffic features could identify stealthy attacks in power system ICS, while more recent works apply NNs to SCADA data streams for detecting anomalies in chemical and water plants Raman et al. (2020), Abdelaty et al. (2021) and Boateng et al. (2022)

- **Instance-based Learning:** These algorithms classify new instances based on their distance to existing instances in the training dataset. They are

considered lazy learning algorithms and have been applied to IDS (Palácios et al. 2013; Muda et al. 2011; Kumar et al. 2013). In ICS, K NEAREST Neighbour (K-NN) has been used for process level datasets (Goh, Adepu, Tan et al. 2017), where abnormal sensor values and control commands could be effectively identified by comparing them with historical normal states. This makes instance-based methods suitable for ICS datasets that display recurring operational patterns.

- **Multinomial LR:** Similar to linear regression but assumes a Bernoulli distribution for the dependent variable. LR uses the logistic function for prediction, generating probabilities to quantify relationships between dependent and independent variables. While LR can benefit from larger feature sets, its efficacy in IDS remains limited (Tsai et al. 2009). In ICS, logistic regression has been employed for binary classification of attack vs. normal behaviour in SCADA traffic. Although its performance is often outperformed by tree-based or deep learning methods, its interpretability provides value in critical infrastructure contexts where transparency of decision-making is important for operators (Hindy et al. 2018; Dev et al. 2024).

2.3.1.2 Decision Tree-Based Approaches

Decision tree algorithms are characterized by their interpretability, representing decision rules as IF-ELSE structures. They construct a tree-like hierarchy, with internal nodes representing tests on features, branches representing test outcomes, and leaves denoting class labels. In ICS, Decision Tree have been used to detect malicious commands and process deviations by learning decision rules from SCADA datasets. For example, Junejo and D. Yau (2016) used Decision Tree to classify cyberattacks targeting process sensors, demonstrating its

ability to capture attack patterns specific to industrial processes.

- **Random Forest (RF):** An ensemble learning technique that builds multiple decision trees using random subsets of features, improving robustness against over-fitting (Breiman 2001). RF models have been widely adopted in ICS due to their robustness against noisy sensor and traffic data. Hink et al. (2014) demonstrated that RFs effectively distinguish between natural disturbances and cyberattacks in power grid ICS, highlighting their suitability for mixed physical and network-level data.

- **Applications in IDS:** Decision tree algorithms have shown success in network-level IDS ((Sahu et al. 2015; Hasan et al. 2014)). Ensemble methods like AdaBoost and XGBoost have also been employed for intrusion detection in IoT networks (Moustafa et al. 2018), in ICS security, ensemble methods such as AdaBoost and XGBoost have been shown to outperform single classifiers by combining weak learners across diverse SCADA datasets. For instance, Kravchik et al. (2018) applied boosting-based ensembles on process data, achieving high accuracy in detecting cyber attacks.

2.3.1.3 Generative Approaches

Generative approaches predict class membership based on the probabilities of an object belonging to a particular class.

- **Bayesian Networks (BayesNet) and Naive Bayes (NB):** These Bayesian classifiers are commonly used in IDS (Koc et al. 2012; Xiao et al. 2014). NB assumes attribute independence given the class value, making it scalable for high-dimensional data. BayesNets, represented as directed acyc-

lic graphs, capture dependencies between variables and are more suitable when attributes exhibit interdependencies (Friedman et al. 1997).

- **Applications:** Bayesian networks have been used with Radio Tomographic Imaging (RTI) for anomaly detection (Q. Lin et al. 2018). In ICS security, Bayesian Networks have been used to model probabilistic dependencies between process variables, SCADA commands, and potential intrusion events. For instance, Q. Zhang et al. (2017) proposed a fuzzy probability Bayesian Network for dynamic cybersecurity risk assessment in ICS, demonstrating its effectiveness in reasoning about uncertainties inherent in both cyber and physical process data. Naive Bayes has been employed alongside other algorithms to enhance virtual machine security (Kumara et al. 2018).In ICS, Naive Bayes has been used for process anomaly detection by learning probability distributions of sensor data. For example, Junejo and D. Yau (2016) applied NB on ICS dataset to classify process attacks, demonstrating that lightweight generative models can still achieve effective detection in resource-constrained industrial environments.

Building upon these methods, V et al. (2025) evaluated ML approaches for cyberattack mitigation in ICS and found that supervised classifiers remain effective but struggle with generalisation across evolving attack types.

2.3.1.4 Deep Learning Based Supervised Learning Approaches

Deep learning, an extension of ML, enables Neural Networks (NNs) to automatically learn complex feature representations from data. This learning techniques has been studied in the anomaly and intrusion detection in ICS (Illy et

al. 2022; Prasanna et al. 2023; Nie et al. 2024).

- **Convolutional Neural Networks (CNNs):** Highly effective in visual image analysis, CNNs address over-fitting issues associated with fully connected networks like MLP. They leverage hierarchical patterns in data to learn complex patterns from simpler ones. CNNs have been used for classifying PLC programs using PMU data (Stockman et al. 2019), keystroke detection using mobile phone sensor data (Giallanza et al. 2019), anomaly detection using thermal imaging (Amrouch et al. 2017), and. In ICS, CNNs have been employed to analyse time-series data and raw SCADA signals. Kravchik et al. (2018) demonstrated that CNNs could detect multivariate anomalies in the SWaT water treatment testbed, outperforming classical ML methods in identifying stealthy cyber-physical attacks.

- **Recurrent Neural Networks (RNN)s):** Designed for sequential data, RNNs possess edges that traverse time steps, enabling them to capture temporal dependencies. They have been used for vehicle cybersecurity (Loukas et al. 2017) and speech recognition (Fernández et al. 2007) and detecting distributed Denial of Services Attacks in IoT-enabled ICS (Varghese et al. 2024). Long Short Term Memory (LSTM), a specialized RNN, has shown promise in intrusion detection, fault management, and anomaly detection in various applications (Zizzo et al. 2019; Ieracitano et al. 2020; Ariharan et al. 2019; Y.-Q. Li et al. 2020; Hussain et al. 2024). In ICS anomaly detection, RNNs and particularly LSTMs are well-suited for modelling sequential process data and control signals Inoue et al. (2017) applied LSTMs on the ICS dataset, showing that they effectively capture temporal dependencies in sensor readings to detect cyberattacks on water treatment processes.

- **Autoencoder:** Supervised Autoencoders are often trained with labelled normal and attack samples, where the reconstruction error is used as a discriminative feature to classify inputs as benign or malicious. This allows autoencoders to capture complex feature representations of ICS data streams, improving the detection accuracy of known attack types. Autoencoders have been widely adopted for ICS intrusion detection due to their ability to learn compact representations of normal operation. Gauthama Raman et al. (2020) applied deep autoencoders to SCADA data from a water treatment plant, successfully identifying anomalies linked to cyberattacks.

2.3.2 Unsupervised ML Techniques

This section explores various unsupervised machine learning techniques and their applications in anomaly detection, particularly within the context of ICS and cybersecurity.

2.3.2.1 Centroid-Based Clustering

This technique revolves around central vectors, not necessarily belonging to the dataset, that define cluster centres. Data points are assigned to clusters based on their proximity to these centres. K-means, a popular centroid-based algorithm, requires predefining the number of clusters (k). Several studies highlight its applications:

- **K-means and variants:** These methods, are widely used in ICS intrusion detection because of their efficiency in grouping process and net-

work data without requiring labelled datasets. These approaches are well-suited for SCADA and ICS datasets, where distinct operational states can be clustered to identify abnormal behaviour. For example, Bhattacharjee et al. (2018) applied K-means clustering to detect compromised smart meters in Advanced Metering Infrastructure (AMI), while Demertzis et al. (2020) used fuzzy C-means to develop a resilient intrusion detection system for critical infrastructures. Similarly, Alves et al. (2018) combined K-means with Local Outlier Factor (LOF) to enhance PLC security in ICS environments. In wireless ICS contexts, W. Liu et al. (2014) demonstrated how K-means applied to channel state information could distinguish between legitimate and malicious users. These studies highlight the adaptability of centroid-based clustering to ICS-specific anomaly detection challenges. furthermore, Ruslan et al. (2025) extended clustering-based strategies to review AI-driven clustering protocols for IoT and IIoT environments which showed their potential for unsupervised anomaly detection in ICS where labelled data is scarce.

2.3.2.2 Distribution-Based Clustering

This statistical technique groups objects based on their underlying probability distributions, making it suitable for modelling uncertainty in complex datasets. However, it is prone to over-fitting if model complexity is not carefully managed.

- **Gaussian Mixture Models:** In ICS, GMMs have been applied to capture the probabilistic behaviour of process and SCADA data. For instance, Kiss, Bela Genge et al. (2015) applied GMMs to PLC sensor data to detect stealthy cyberattacks, while Y. Zhang et al. (2011) demonstrated their effectiveness in identifying false data injection in smart grid networks.

These studies highlight GMMs as powerful tools for anomaly detection in ICS, outperforming traditional k-means clustering in scenarios involving subtle attacks.

2.3.2.3 Density-Based Clustering

This approach focuses on separating regions of high data density from sparser areas, aiding in noise reduction and decision boundary establishment, making it effective for anomaly detection in noisy environments.

- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN):** In ICS, DBSCAN and its variants have been widely used to detect cyberattacks in SCADA and smart grid systems. For example, Almalawi, Fahad et al. (2016) applied DBSCAN to detect intrusions in SCADA systems, showing superior accuracy compared to k-means. Similarly, Otoum et al. (2017) proposed an enhanced DBSCAN to mitigate false negatives in smart grid intrusion detection, while Sharma et al. (2023) combined PCA and DBSCAN to identify false data injection attacks in power grid ICS. Moreover, Çelik et al. (2011) utilised DBSCAN, for anomaly detection on temperature data and Abid et al. (2017) proposed use of DBSCAN Outlier Detection (DBSCAN-OD), a DBSCAN variant, for outlier detection in noisy environments. These works underscore DBSCAN's utility in distinguishing between normal operations and malicious activity in industrial environments.

2.3.2.4 Hierarchical Clustering

Also known as connectivity-based clustering, this method establishes a hierarchical structure of clusters. It encompasses two primary categories, **Agglomerative Clustering** or bottom-up or **Divisive Clustering** or top-down, making it suitable for datasets where hierarchical structures exists.

- In ICS security, hierarchical clustering has been applied to group anomalies in both network traffic and process data. For example, Ghaeini et al. (2016) introduced HAMIDS, a hierarchical intrusion detection system for ICS, where distributed anomaly detectors are structured in a layered fashion. Similarly, Bukharev et al. (2022) applied hierarchical cluster analysis to ICS data exposed to cyberattacks, enabling detection of abnormal states in operator stations and SCADA traffic. Ren et al. (2018) further demonstrated a multi-level approach (EDMAND) for anomaly detection in SCADA networks, using hierarchical grouping of anomalies across communication layers.

2.3.2.5 Association Rule Mining (ARM)

ARM uncovers relationships within datasets by identifying frequent patterns and correlations between features. This characteristic makes it suitable for uncovering hidden relationships in operational data.It has been applied in IDS for detecting frequent attack patterns and invariants in both IT and ICS contexts

- **ARM:** In ICS, ARM has been used to reveal critical states and invariants that help distinguish normal and anomalous process behaviour. For example, Khalili et al. (2015) applied the Apriori algorithm to identify critical system states, incorporating expert knowledge to improve intrusion detection. Pal et al. (2017) explored ARM for generating invariants in

a water treatment plant (SWaT), with subsequent extensions by Umer et al. (2017) and Umer et al. (2020) and Mujeeb Ahmed et al. (2021) using FP-Growth for more scalable invariant mining in real-world ICS. Recent research further strengthens the role of ARM in ICS intrusion detection. Mehmood et al. (2024) applied ARM in the generation of synthetic cyber-physical attack samples to improve IDS training, while Samiah et al. (2025)combined ARM with decision tree methods to extract process invariants for anomaly detection in ICS. Together, these works highlight ARM's evolving role as both a foundational and modern approach for uncovering structured, interpretable knowledge in ICS security.

2.3.2.6 Deep Learning Based Unsupervised Learning Approaches

Unsupervised deep learning techniques such as autoencoders, Generative Adversarial Network (GAN)s, and Deep Belief Networks (DBN)s learn data representations without requiring labelled training data. This makes them highly attractive for ICS intrusion detection, where labelled attack datasets are scarce. These methods can capture complex temporal and spatial correlations in both network traffic and process data, enabling the identification of subtle and previously unseen cyberattacks.

- **Autoencoder-Based Deep Clustering:** Autoencoders are widely used for representation learning, compressing input data into a lower-dimensional latent space that preserves essential features. This latent space is then clustered to identify anomalies or hidden patterns. Several frameworks extend this principle: Deep Embedding Network (DEN) jointly optimizes reconstruction and clustering objectives; Deep Clustering Network (DCN) integrates k-means with deep representation learning, Deep Embedded Regularised Clustering (DEPICT) improves stability through regulariza-

tion and Deep Continuous Clustering (DCC) applies convolutional autoencoders to image-like data structures. These approaches have proven effective in extracting robust embeddings for anomaly detection tasks. In ICS contexts, autoencoder-based clustering has been applied to both process data and SCADA traffic. Inoue et al. (2017) demonstrated stacked autoencoders on the SWaT water treatment testbed for detecting false data injection attacks. More recent work such as Aslam et al. (2024) refined autoencoder-based anomaly detection for SCADA networks, reducing false alarms, while Ruan et al. (2023) highlighted the utility of variational autoencoderss in smart grid cybersecurity. Collectively, these works show that autoencoder-based clustering methods (DEN, DCN, DEPICT, DCC) can be effectively adapted to ICS data, enabling unsupervised detection of attacks and operational anomalies.

- **GAN-Based Deep Clustering:** GANs were originally proposed for data generation, with a generator and discriminator competing to model complex distributions. More recently, GANs have been adapted for deep clustering, where the latent features learned by the discriminator can be partitioned into meaningful clusters. This makes GAN-based clustering attractive for ICS intrusion detection, where labelled attack data is scarce and operational data is high-dimensional.

  In ICS, Perales Gómez et al. (2020) proposed MADICS, a GAN-driven anomaly detection methodology that combined clustering analysis with GAN representations to uncover hidden structures in ICS datasets. Bedeuro Kim et al. (2023) evaluated MAD-GAN on widely used testbeds such as SWaT and WADI, showing how GAN-based clustering separates normal operational states from abnormal attack-induced states. Building on this, J.-R. Jiang et al. (2022) applied GAN-based clustering to SCADA network

traffic, successfully distinguishing between normal and attack flows without prior labels. More recent frameworks, such as W. Shen et al. (2024) with CluSAD, integrated self-supervised learning and GANs for SCADA-based industrial IoT, reinforcing the effectiveness of GAN latent feature clustering in identifying anomalies across heterogeneous ICS data sources.
Collectively, these studies highlight GAN-based clustering as a promising unsupervised learning approach for ICS anomaly detection, leveraging adversarially trained latent spaces to improve detection of stealthy attacks in SCADA and smart grid environments.

2.3.3 Semi-Supervised Learning

Semi-supervised learning combines limited labelled data with a larger pool of unlabelled data, offering a middle ground between supervised and unsupervised approaches. This is particularly relevant for intrusion detection in ICS, where obtaining labelled attack data is costly, and in many cases impractical, due to operational and security constraints. Early studies demonstrated the feasibility of applying semi-supervised methods in intrusion detection, laying the groundwork for their adoption for example Maglaras, J. Jiang and Cruz (2014) introduced a distributed IDS composed of a cluster of OCSVM models that can precisely discern the origin and timing of an attack. Each model was trained on a network traffic segment divided by packet source. Similarly Huda, Abawajy et al. (2019) created a semi-supervised malware detection model, utilizing data derived from both static and dynamic malware features. They showed that this model outperforms its supervised equivalent in identifying new malware.
Recent contributions highlight growing importance of Semi-Supervised Learning for ICS anomaly detection. Joshi et al. (2020) applied a semi-supervised

approach to SCADA attack detection in a gas pipeline system, successfully combining scarce labelled data with abundant normal traffic. Qi et al. (2021) introduced a deep representation learning framework for smart grids, using semi-supervised anomaly detection to capture cyberattacks in SCADA data streams. Similarly, Loo et al. (2023) presented a semi-supervised detection method for water storage ICS, confirming its capability against cyber-physical attacks.
Beyond single ICS domains, new frameworks extend to cross-domain and federated semi-supervised learning. Chen et al. (2022) developed a cross-domain semi-supervised model to handle imbalanced ICS traffic, improving detection generalisation. Aouedi et al. (2022) designed a federated semi-supervised learning architecture for industrial IoT and SCADA, reducing reliance on centralised datasets. **perales2023interpretable** further advanced this by introducing an interpretable semi-supervised anomaly detection framework, enabling transparent decision-making in industrial contexts.
These works demonstrate that semi-supervised learning reduces reliance on fully labelled datasets while enhancing resilience against evolving attacks in ICS and SCADA environments.

2.3.4 Reinforcement Learning (RL)

Reinforcement Learning (RL), distinct from other machine learning techniques, centres on an agent learning through interaction with an environment. The core components of an RL system are the agent, the environment, and the reward signal. The agent takes actions within the environment and receives rewards, either positive or negative, based on the outcomes of those actions. This reward-driven feedback loop drives the agent's learning process. Unlike other machine learning paradigms, RL does not necessitate a pre-existing dataset for training. Several prominent RL algorithms include:

1. **Temporal Difference Learning (TD Learning):** A model-free algorithm that utilizes bootstrapping to train a model based on the current estimate of the value function. TD learning samples experiences from the environment and updates its estimates accordingly (Barto et al. 1997). Qiao et al. (2024) has studied the integration of advantage actor-critic reinforcement learning with a long short-term memory network in improving security of traffic signal control against last vehicle attack.

2. **State-Action-Reward-State-Action (SARSA):** An on-policy algorithm that learns a policy by interacting with the environment and updating its estimates based on the rewards received for taking specific actions in specific states.

3. **Q-Learning:** A model-free algorithm that enables an agent to learn an optimal policy for maximizing rewards. Q-learning does not require a model of the environment and can handle stochastic transitions and rewards. Sangoleye et al. (2024) studied the application of various RL models including Q-Network for network intrusion detection in ICS.

RL has demonstrated its utility in various security applications. Barto et al. (1997) employed RL for intrusion detection in a simulated Wireless Sensor Network (WSN), showcasing its superior performance compared to adaptive machine learning-based IDS. Kurt et al. (2018) proposed a model-free RL approach for anomaly detection in smart grids, addressing the Partially Observable Markov Decision Process (POMDP) problem. In the context of Cyber-Physical System (CPS) security, Feng et al. (2017) formulated the defence problem as a two-player zero-sum game, leveraging deep RL to optimize an actor-critic neural network architecture. Similarly, Panfili et al. (2018) modelled the attack problem as a multi-agent general-sum game, employing RL to determine

optimal prevention actions and associated costs. J. Yan et al. (2016) investigated Q-learning for vulnerability assessment in smart grids, focusing on sequential topological attacks. Their findings, based on simulations of IEEE 5-bus, RTS-79, and IEEE 300-bus systems, highlighted the potential of Q-learning to identify vulnerabilities. RL has also proven valuable for anomaly detection in Unmanned Aerial Vehicle (UAV)s. Lu et al. (2017) utilised RL to detect anomalous motor behaviour in UAVs by monitoring motor temperature using sensors and a Raspberry Pi-based processing unit.

2.3.5 Recent Advances: Digital Twins, Preventive Maintenance, and IDS

Industrial informatics research shows Digital Twin technology functions as an integrated system which enhances operational resilience and cybersecurity protection for ICS. A digital twin creates an instant virtual model of physical operations which allows predictive analysis for system performance under typical and unusual operating states.

The adoption of Digital Twins continues to grow because they enhance both cybersecurity system capabilities and their ability to detect intrusions. A. Singh (2024) analysed the main obstacles to secure Digital Twins implementation in ICS systems and showed how cloud integration and physical-virtual system data exchange operations lead to security risks. The authors Oyedotun et al. (2025) demonstrated through their research how Digital Twins-based modelling identifies system anomalies in SCADA and DCS systems through virtual system activity to intrusion behaviour connections. Larsson et al. (2025) built upon this research by implementing homomorphic encryption for Digital Twins environments which protects industrial automation data during processing.

These studies highlight the emerging consensus that Digital Twin can enhance the detection and mitigation of cyber-attacks by embedding security monitoring within virtual replicas of physical assets.

Digital Twins function as a solution for industrial operations to perform predictive and preventive maintenance tasks. Qin et al. (2026) conducted a thorough evaluation of robot Digital Twins which proved their ability to identify component failures before they happen and perform maintenance tasks at peak operational efficiency. Anbalagan et al. (2025) presented a lightweight CNC digital process twin system which combined IIoT with OPC UA for real-time equipment monitoring to enable early fault detection and preventive maintenance. Urrea (2025) positions Digital Twins as part of Industry 5.0 framework which requires predictive maintenance and cybersecurity to build sustainable and resilient robotics systems.

The majority of current studies focus on cybersecurity and preventive maintenance independently yet researchers now show growing interest in their combined approach. Digital twins enables real-time process data access which helps predicting equipment failures and tracking system operational performance. Yet, while process data captures what is happening within the system, it does not reveal who is initiating the action or how it is being executed.Network traffic provides better contextual attribution because it functions as the intent channel which reveals both external system interactions and attempted intrusions. Process data serves as an essential requirement for detecting abnormal states and operational reliability but it does not provide the detailed information which network-level insights deliver. Integrating both sources — process state information and network traffic — is thus critical for achieving real-time detection of cyber threats and for developing holistic ICS security solutions.

2.3.6 Learning Using Privileged Information (LUPI)

The LUPI paradigm (Vapnik and Vashist 2009b) enables ML models to use training-specific auxiliary knowledge for better generalization. The LUPI paradigm has been used in cybersecurity for botnet detection (Sapello et al. 2017), malware classification (Shaikh et al. 2020), and anomaly detection (Celik et al. 2018). However, its adoption in ICS remains minimal. The main limitations of this approach are its dependence on well-structured privileged information and the difficulty of aligning different data modalities during training. This research extends the LUPI approach by investigating the use of process data as privileged information to enhance IDS performance for ICS.

2.4 Summary

In this chapter I delved into the world of ICS, their security challenges and different type of intrusion detection Systems applicable to ICS domain. I have also reviewed the existing literature on the application of Machine Learning in ICS Intrusion Detection Systems identifying their limitations and research gap.
ICS are vital to critical infrastructure, managing processes in in sectors like power, transportation, and manufacturing. Unlike traditional IT systems, ICS prioritize real-time operation and data integrity, relying on specific control algorithms and communication protocols. This reliance on precise functioning and interconnectedness makes ICS vulnerable to cyberattacks, potentially leading to physical damage and safety risks. IDS are crucial for identifying malicious activities within ICS. Traditional IDS, often used in IT environments, struggle to address the unique challenges posed by ICS. These challenges include:

- **Real-time operation:** Continuous monitoring without disrupting critical processes.
- **Limited resources:** Constraints in computing power and storage capacity.
- **Unique protocols:** Reliance on specialized communication protocols.
- **Semantic attacks:** Exploiting control systems and physical processes without triggering traditional security alerts.

Machine Learning (ML) for ICS IDS offers promising solutions for addressing these challenges. While the key applications of ML in ICS IDS can be summarised in **Protocol Analysis** (detecting deviations from expected communication patterns defined by industrial protocols like MODBUS, DNP3, and ICCP) (e.g. (Yusheng et al. 2017; Y. Yang et al. 2013; Hong et al. 2014; Curtis 2005)), **Traffic Mining** (identifying abnormal traffic patterns within ICS networks using techniques like clustering, association rule mining, and neural networks (e,g, (Hou et al. 2012; Linda, Vollmer et al. 2009; Ashfaq et al. 2017; Javaid et al. 2016; Ferling et al. 2018; Dong et al. 2018)) and **Control Process Analysis** (analysing the behaviour of control systems and physical processes to detect anomalies, including response injection attacks and stealthy attacks that manipulate sensor data (e.g. (Carcano, Coletta et al. 2011; Hadžiosmanović et al. 2014; Colbert et al. 2016; Moya et al. 2018; Urbina et al. 2016; Kleinmann et al. 2018; Tian et al. 2018)), the efficacy of different ML techniques is investigated in enhancing intrusion detection and overall security posture of ICS: **Supervised Learning:** Utilises labelled data of both normal and malicious behaviour to train models for accurate attack detection (e.g. (Kwon et al. 2015; Ahmad et al. 2014; Al-Jarrah et al. 2015; Moustafa et al. 2018; Kumara et al. 2018; Stockman et al. 2019; Ieracitano et al. 2020; Ariharan et al. 2019)). However, it requires

comprehensive attack datasets, which can be limited, and may not effectively identify unknown attacks. **Unsupervised Learning:** Relies on identifying anomalies in normal ICS behaviour, eliminating the need for labelled attack data and enabling the detection of zero-day attacks (e.g. (Bhattacharjee et al. 2018; Umer et al. 2020; Mujeeb Ahmed et al. 2021; Shinde et al. 2018)). However, it can lead to a higher rate of false alarms. **Semi-Supervised Learning:** Combines labelled and unlabelled data, leveraging the strengths of both supervised and unsupervised learning (e.g.
(Huda, Abawajy et al. 2019; Huda, Miah et al. 2017). This approach shows promise in addressing the limitations of using solely labelled or unlabelled data. **Reinforcement Learning:** Focuses on an agent learning through interaction with the ICS environment, making decisions based on rewards and penalties(Kurt et al. 2018; Panfili et al. 2018; Lu et al. 2017)). Multiple new approaches have emerged through recent advancements. Notably, digital twin technology is emerging as a platform for both cybersecurity monitoring and preventive maintenance in ICS. A. Singh (2024) along with Oyedotun et al. (2025) and Larsson et al. (2025) showed that Digital Twins enable the integration of security analytics within virtual process replicas. In parallel, Qin et al. (2026) and Anbalagan et al. (2025) and Urrea (2025) demonstrated the effectiveness of Digital Twins for predictive maintenance and operational resilience. The two research paths function separately but show that digital twins achieve operational reliability and cybersecurity monitoring through shared process data integration.
Scientists conduct individual studies about cybersecurity and preventive maintenance but research on their combined approach is becoming more popular. Users can predict physical system failures and track process conditions through the shared real-time process data access provided by digital twins. Yet, while

process data captures what is happening within the system, it does not reveal who is initiating the action or how it is being executed.Network traffic enables better contextual attribution because it serves as the intent channel which shows both external interactions and attempted intrusions. Process data continues to play a vital role in detecting abnormal states and maintaining operational reliability yet network-level information provides more comprehensive explanations about activities within the network. Integrating both sources — process state information and network traffic — is thus critical for detection of cyber threats and for developing holistic ICS security solutions.
The three **research gaps** identified in the existing body of research looking at IDS for ICS include:

- Most IDS approaches for ICS still rely predominantly on network traffic data, with limited integration of process-level insights.
- The problem of detecting and differentiating multi-attack scenarios remains unexplored, even though such attacks reflect realistic adversarial behaviours.
- The application of digital twins as emerging paradigms exists independently for predictive maintenance and cybersecurity but researchers have not shown their combined use for developing complete intrusion detection systems.

The research aims to reduce these knowledge gaps through its proposed framework which uses LUPI and process information to improve detection accuracy by integrating network and process data. The framework enables ICS cybersecurity advantages because it addresses various attack scenarios by using process state knowledge which align with current industry standards. Figure 2.2 gives a clear, visual overview of IDS approaches for ICS, summarising

the literature discussed in this chapter.

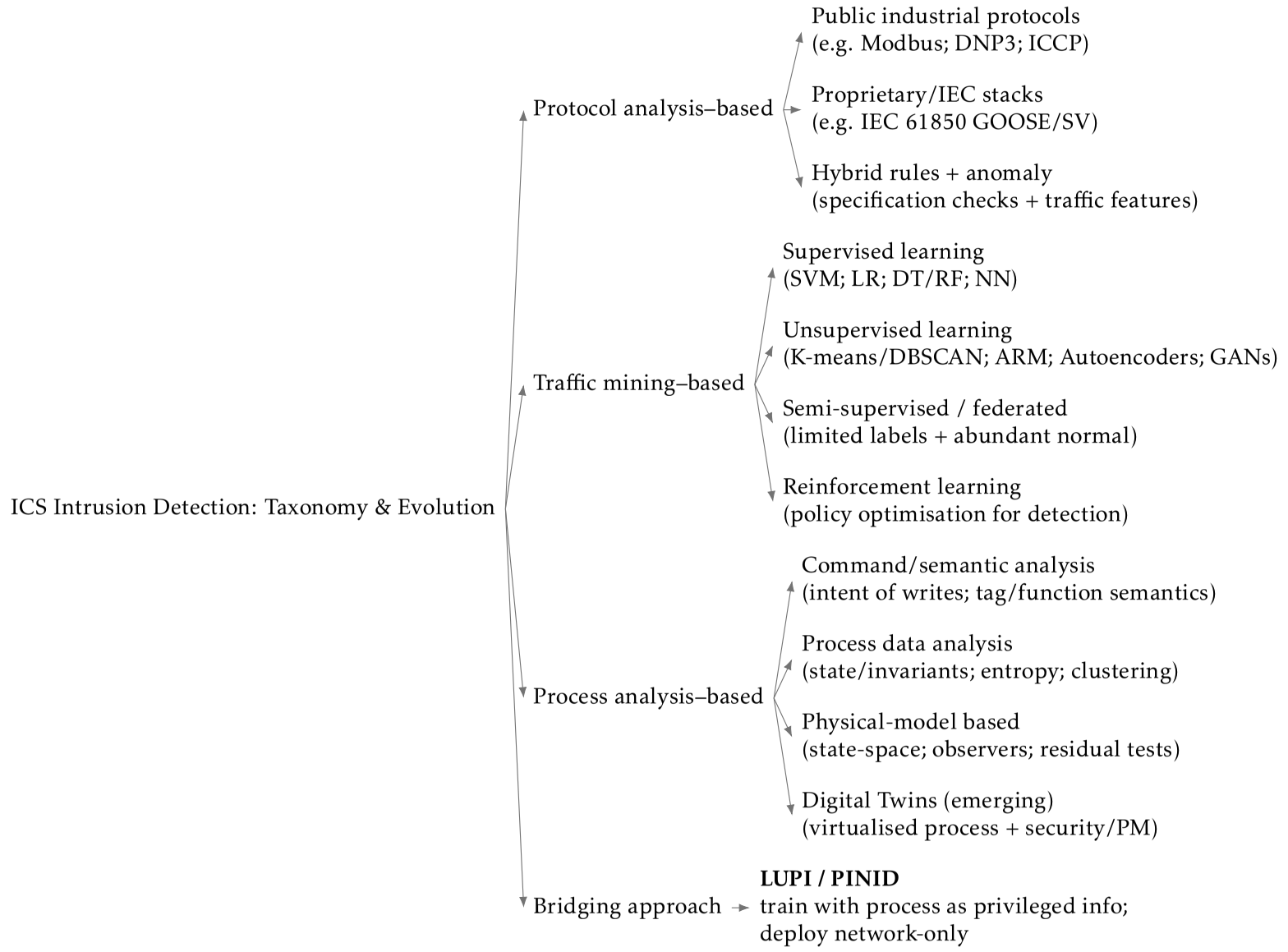

Figure 2.2: Taxonomy of ICS intrusion detection approaches and recent directions, situating this thesis (LUPI/PINID) as a bridge between traffic-centric and process-aware methods.

Chapter 3

Materials and methodology

3.1 Dataset selection

We required a dataset on an experimental ICS that follows a hierarchical (Purdue-style) network segmentation, and that captures both network traffic and historian (process) data under the same operating conditions during attacks. Because our framework is supervised, we also require reliable labels. We deliberately avoid fully simulated or synthetic datasets. They are useful for benchmarks, but they often miss real-world timing, device behaviour, and messy protocol behaviour—making models look better than they really are and hurting generalisability.(Conti et al. 2021; Cordero et al. 2019; Dehlaghi-Ghadim et al. 2023). Among public datasets, SWaT stands out: it gives both historian CSVs and raw PCAPs captured on a real water-treatment testbed, with attacks run while both data type were recorded and labelled which meets all our criteria (iTrust, SUTD 2016; Aditya P. Mathur et al. 2016b).

Table 3.1 summarises prominent ICS datasets against our criteria. Only SWaT fully meets our present needs; others are noted as Partial (e.g. process-only or network-only) or Not suitable (simulated).

Dataset (year)	Domain	Testbed type	Purdue	Proc+Net	Sync (attack)	Labels	Notes / Sources
SWaT (2015)	Water treatment	**Physical**	✓	✓	✓	✓	Provides historian CSVs and PCAPs from same runs with attack windows; widely reused. (iTrust, SUTD 2016; Aditya P. Mathur et al. 2016b)
WADI (2017)	Water distribution	Physical	✓	–	–	✓	Public process logs; network traces not routinely released—may require request. (iTrust, SUTD 2017; Turrin et al. 2020)
HAI 21.03 (2021)	Multi-plant (HIL)	**HIL hybrid**	✓	✗	✓	✓	Operational time series; no public PCAP; modern attacks; good for future cross-validation. (Shin et al. 2021; Bongjun Kim et al. 2023)
MSU Gas Pipeline (2014–2015)	Gas pipeline	Physical	✓	–	–	✓	Labeled ICS *network* datasets; limited or no synchronised historian logs. (Morris and W. Gao 2014; IMPACT Cyber Trust 2015; Turnipseed 2015)
WDT (2021)	Water distribution	**HIL hybrid**	✓	✓	✓	✓	PCAP + physical variables via IEEE DataPort; hybrid hydraulics. (Faramondi et al. 2021)
PowerDuck (2022)	Substation (GOOSE)	Physical	✓	–	✓	✓	Real substation network traces with labeled attacker packets; no process historian. (Zemanek et al. 2022)
BATADAL 2.0 (2024)	Water distribution	**Simulated**	✓	✓	✓	✓	Adds network traces via simulation; not suitable given our no-simulation requirement. (Critical Infrastructure Systems Lab 2024)
ICS-Flow (2023)	Generic ICS	**Sim/emulated**	✓	✓	✓	✓	Both streams but simulated/emulated; useful for methods, not for primary evaluation. (Dehlaghi-Ghadim et al. 2023)
HVAC Campus (2019)	Building automation	Real deployment	✓	–	–	✗	Packet-header traces from a live HVAC; limited labels. (Ndonda et al. 2019)

Table 3.1: Public ICS datasets vs. our selection criteria (real testbed, Purdue-style hierarchy, synchronised network & historian during attacks, and labels).

The comparison in Table 3.1 motivates our choice of the SWaT testbed. It is the only public option that meets all four requirements—collection on a physical ICS with Purdue-style segmentation, synchronised historian and network captures under the same attack conditions, and reliable labels—so it enables us to train and validate a supervised, process-informed NIDS under realistic constraints. We therefore use SWaT to evaluate the proposed framework's ability to improve NIDS performance in an ICS setting by integrating process and network data during training and assessing effectiveness on held-out sequences, while keeping the deployment-time detector network-only. (iTrust, SUTD 2016; Aditya P. Mathur et al. 2016b)

3.2 SWaT Test bed and data

The Secure Water Treatment (SWaT) test bed , owned by the Singapore University of Technology (Goh, Adepu, Junejo et al. 2016), is a diminutive water treatment plant comprising six process stages which can effectively deliver 5 gal/minutes of double-filtered water. The SWaT network architecture depicted in Figure 3.1 exhibits a distributed control system structure in which a redundant pair of programmable logic controller (PLC) manages the operation of each stage of the process.

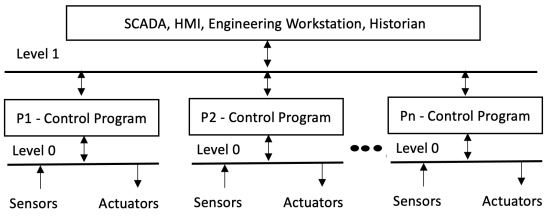

Figure 3.1: Overview of SWaT Network Architecture, adapted from (Goh, Adepu, Junejo et al. 2016)

Data: Although SWaT was collected in 2015, it remains one of the few public datasets offering aligned process and network streams. We acknowledge limitations (legacy protocols, a fixed set of scripted attacks), but the dataset supports reproducible baselines and comparison with contemporary work (Al-Dhaheri et al. 2022). A variety of datasets are available to be accessed from the SWaT test-bed upon request. this research was conducted utilising the SWaT A1& A2-Dec 2015 datasets. The datasets include the recordings of the network traffic and the physical status of 51 field instruments during 11 consecutive days of continuous operation. Over the initial week of operation, data was collected from standard activities, while the last four days incorporated data from 41 attacks. According to Goh, Adepu, Junejo et al. (2016), by exploiting the Level 1 communication link presented in Figure3.1 attacks were initiated and the data in the application layer of the communication packets were maliciously manipulated before being injected back to the PLCs. Each of the field instrument's physical features were monitored and recorded as process data in the Historian server at set spans of one second. Network data was taken from Level 1 shown in Figure3.1 with a much greater frequency. It has been suggested that network data only contains data that is beneficial for intrusion detection (ibid.). All process and network data that is acquired is marked with timestamp, allowing the database provider to pinpoint any data created during an attack.

After integrating the network and historian streams, the dataset used in this study contains **495,000** records (network–process pairs). We use a **70/30** train/test split, yielding **346,500** training and **148,500** test records; process data are used as privileged information during training only, while the IDS model use network data at runtime. (see section 3.4).

3.2.1 Network Data Collection

The SWaT dataset comprises of network information that was obtained via the interaction of the Supervisory Control and Data Acquisition (SCADA) and PLCs. Network traffic was logged at a high rate of milliseconds. The dataset includes 19 selected features that are deemed to be of worth for intrusion detection. These features are *Date*, *Time*, *Origin*, *Type*, *Interface Name*, *Interface Direction*, *Source IP*, *Destination IP*, *Protocol*, *Proxy Source IP*, *Application Name*, *Modbus Function Code*, *Modbus Function Description*, *Modbus Transaction ID*, *SCADA Tag*, *Modbus Value(the payload of the Modbus protocol consists of multiple registers)*, *Service/Destination Port*, *Source Port*.

For our experimentation, we have incorporated the request and response of the same transaction into a single record, taking into account the transaction ID. Initially, when examining the Modbus payload, we transferred the hex-encoded binary numbers into floating point values and obtained values that were representative to the SCADA tag obtainable within the dataset. In this context, it was the beginning value of the Modbus value field. Table 3.2 presents a record of the Network data used in this study.

Time: 10:37:20 **Label:** Attack

Modbus_Function_Code = 76	Modbus_Transaction_ID = 37889	Request_NumberofElements = 1
Modbus_Value = 848.12866	service = 44818	s_port = 52544
orig_192.168.1.48 = 1	type_log = 1	i/f_name_eth1 = 1
i/f_dir_outbound = 1	src_192.168.1.10 = 0	src_192.168.1.20 = 0
src_192.168.1.30 = 1	src_192.168.1.60 = 0	dst_192.168.1.10 = 0
dst_192.168.1.20 = 0	dst_192.168.1.30 = 0	dst_192.168.1.40 = 1
proto_tcp = 1	appi_name_CIP_read_tag_service = 1	proxy_src_ip_192.168.1.10 = 0
proxy_src_ip_192.168.1.20 = 0	proxy_src_ip_192.168.1.30 = 1	proxy_src_ip_192.168.1.60 = 0
Modbus_Response_Function_1 = 1	Modbus_Request_Function_1 = 1	SCADA_Tag_HMI_AIT202 = 0
SCADA_Tag_HMI_FIT201 = 0	SCADA_Tag_HMI_LIT101 = 0	SCADA_Tag_HMI_LIT301 = 0
SCADA_Tag_HMI_LIT401 = 1		

Note. Underscored field names are shown verbatim from the dataset header. Binary flags (0/1) indicate presence for the corresponding source/destination or proxy field; ports are in integer form.

Table 3.2: Example SWaT network record (compact three-column view).

3.2.2 Historian Data Collection

The SWaT dataset comes with two versions of process data from the normal operation of the plant presented by (Goh, Adepu, Junejo et al. 2016). In the first version, data recording started at the draining stage of the storage tank which took 30 minutes. This stage is part of the maintenance process and is out of the normal operation phase. Therefore a second version was presented by removing the first 30 minutes of the recorded data. in our experiments, we have used the latter version which includes data from 51 field instruments. In Industrial plants, generally, process data are reported periodically to the Historian server at a fixed rate which is dependant on the characteristics and configuration of the process parameters. In SWaT, dataset process data are reported to the Historian server every second. Table 3.3 present a sample record from Historian data.

Time: 10:37:20		**Label:** Attack					
FIT101	2.484707	LIT101	863.9532	MV101	2	P101	2
P102	1	AIT201	262.3366	AIT202	8.394835	AIT203	330.2999
FIT201	2.456541	MV201	2	P201	1	P202	1
P203	2	P204	1	P205	2	P206	1
DPIT301	19.75799	FIT301	2.20645	LIT301	810.5294	MV301	1
MV302	2	MV303	1	MV304	1	P301	1
P302	2	AIT401	148.808	AIT402	155.0884	FIT401	1.717874
LIT401	848.3209	P401	1	P402	2	P403	1
P404	1	UV401	2	AIT501	7.873815	AIT502	144.2194
AIT503	261.7278	AIT504	12.15073	FIT501	1.724686	FIT502	1.254259
FIT503	0.7352687	FIT504	0.306761	P501	2	P502	1
PIT501	250.4646	PIT502	1.69801	PIT503	189.1182	FIT601	0.000128152
P601	1	P602	1	P603	1		

Note. Tags follow SWaT naming (e.g. **LIT**=Level Indicator Transmitter, **FIT**=Flow Indicator Transmitter, **AIT**=Analyser Indicator Transmitter, **P**= Pump state, **MV**= Motorized Valve, **UV**= Ultraviolet unit). Actuator states are integer-coded (e.g. 1/2).

Table 3.3: Example SWaT process record (one timestamp).

3.2.3 Attack Types in SWaT

The SWaT dataset includes 41 attack scenarios, 5 of which are attacks with no physical impacts. Since the focus of this research is on attacks aiming at impairing process control, we have excluded these attacks from the dataset. The remaining 36 attacks have been categorised into 4 groups based on the location of the the attack point in the process stage and the number of compromised elements in each attack:

Single Stage Single Point(SSSP): Attacks targeting one point in a single stage or sub-process.

Single Stage Multi-Point (SSMP): Attacks targeting two or more attack points in a single stage or sub-process.

Multi-Stage Single Point (MSSP): Attacks targeting one point in multiple stages or sub-processes.

Multi-Stage Multi-Point (MSMP): Attacks targeting two or more points in multiple stages or sub-processes.

In this dataset Attacks have also been listed based on their physical impact in the physical process.

Attack with physical change: In these attacks actuators have been compromised. Therefore, it changed the physical state of the actuator and the process.

Attack without physical change: In theses attacks the readings of the sensors has been altered. Therefore, the attack had no physical impact on the process.

3.3 Justification of Attacks

We treat an attack scenario as a clearly bounded period where an attacker changes actuator/sensor states or the control logic. In line with Impair Process Control (TA0106), we focus on Brute Force I/O (T0806) and Unauthorised Command Message (T0855) (MITRE Corporation 2025h; MITRE Corporation 2025a; MITRE Corporation 2025n).

Why multi-attack in ICS is not just data mining

ICS are cyber–physical systems: a CIP/Modbus request only matters in terms of what it should do to the plant (actuators and sensors). If we judge packets one by one (no sequence model), a multi-attack campaign can still look like normal traffic unless we read each command against plant physics, engineering limits, and interlocks (Stouffer et al. 2023). This breaks two common data-mining assumptions: (i) rows are independent and identically distributed (i.i.d.), and (ii) features and labels are purely cyber with no physical meaning (Murphy 2022; Holdbrook et al. 2024). In ICS, a command has a physical effect; "normal" is constrained by control logic and safety functions; and many attacks try to change the process itself rather than just the packet statistics (MITRE Corporation 2025h).

How this approach differs

We developed a NIDS, but we train it to be process-aware. During training, each network record is aligned with its matching historian record and used as privileged information. This teaches the model what a legitimate command should do to the plant. At test time, we used network features only.

We also encode per-record semantics (function code, target tag, requested

value/limits). The detector then flags command records whose implied process effect is inconsistent with normal operation—without needing long sequence models.

Our scenarios focus on Impair Process Control (TA0106), especially Brute Force I/O (T0806) and Unauthorised Command Message (T0855). These often look legitimate at the packet level unless judged against process expectations (Stouffer et al. 2023; MITRE Corporation 2025a; MITRE Corporation 2025n).

Selected attack scenarios: coverage and scale

We chose five attack scenarios to cover both what is changed and how it is carried out. They span the four common shapes—single-source/single-point (SSSP), single-source/multi-point (SSMP), multi-source/single-point (MSSP), and multi-source/multi-point (MSMP)—so the model sees everything from a simple one-actuator manipulations to a coordinated, multi-actuator move. This is deliberate: under Impair Process Control (TA0106), small command tweaks can cause outsized physical effects (MITRE Corporation 2025h). We focus on two technique families: Brute Force I/O (T0806), where an attacker keeps pushing an I/O point toward a target state, and Unauthorised Command Message (T0855), where out-of-policy commands are sent to actuators or controllers (MITRE Corporation 2025a; MITRE Corporation 2025n).

To vary difficulty, SSSP cases (e.g. forcing a single valve or pump) expose clear actuator–sensor mismatches; SSMP and MSSP add cross-unit interactions; MSMP brings timing and coordination, where short command sequences matter as much as individual packets. Because SWaT records synchronised network and historian data for each attack, we can train with process data as privileged information but keep the runtime detection network-only—matching real deployment constraints (iTrust, SUTD 2016; Aditya P. Mathur et al. 2016b). For

scale, Table 3.4 lists the approximate number of merged network transactions per scenario; we also include one "no physical change" case to check that the detector ignores command traffic with no process impact. Overall, this set balances topology and effect and directly targets the two technique families at the heart of our study. The selected attacks are summarised in Table 3.4, and their record counts appear in Table 3.5.

Attack #	Attack Type	Attack Point	Actual Change	Description
1	SSSP	MV-101	Yes	Open MV-101
21	SSMP	LIT-101, MV-101	Yes	Keep MV-101 open; force LIT-101=700
26	MSSP	P-101, LIT-301	Yes	Start P-101; force LIT-301=801
30	MSMP	LIT-101, P-101, MV-201	Yes	Start P-101; open MV-101; force LIT-301=700
36	SSSP	LIT-101	No	Force LIT-101 below low limit

Table 3.4: Subset of attacks used and their scale (approximate merged network records)

Table 3.5 presents the raw per-attack availability (merged network–process pools) with the modelling sample we actually used (equal per-attack, equal Normal), evidencing sparsity and uneven scenario lengths in the source while keeping like-for-like comparison in modelling.

Attack #	Available pool	Share of attack pool	Sampled for modelling	Sampling fraction	Train (70%)	Test (30%)
1	98,185	17.4%	49,500	50.4%	34,650	14,850
21	56,824	10.1%	49,500	87.1%	34,650	14,850
26	188,325	33.4%	49,500	26.3%	34,650	14,850
30	133,970	23.8%	49,500	36.9%	34,650	14,850
36	85,860	15.2%	49,500	57.7%	34,650	14,850
Totals (attacks)		100%	**247,500**	—	**173,250**	**74,250**

. "Available pool" = merged network–historian records per scenario before sampling (sum = 563,164 across the five attacks). "Sampled for modelling" uses an equal per-scenario sample n = 49,500; Normal was sampled to an equal total (247,500).

Table 3.5: Raw per-attack availability vs. modelling sample (network–process pairs).

3.4 Data pre-processing

We prepare the data so features are on the same scale, labels are reliable, and class imbalance doesn't overwhelm training. The steps below match the pipeline we used in our experiments.

3.4.0.1 Normalisation (numeric features).

We standardise all numeric features (process and network) to have mean 0 and variance 1. This puts everything on a common scale, so no single feature dominates just because of its units (Murphy 2022).

3.4.0.2 Encoding (categorical fields).

We encode small categorical fields in the network data (e.g. protocol family, function code) using one-hot indicators. It's a simple, transparent choice that works well when there are only a few possible values (ibid.).

3.4.0.3 Train/test split.

The network dataset is randomly split into **70%** for training and **30%** for testing. The process (historian) records aligned with the training network samples are used as Privileged Information (PI) during training; at test time the network features were used only.

3.4.0.4 Class distribution (modelling dataset) and sampling design

To avoid bias from unequal scenario lengths, we sampled the same number of records per attack scenario and an equal total from Normal, yielding a balanced modelling dataset of **495,000** records (Attack 247,500; Normal 247,500). With a **70/30** split, train and test contain **173,250** vs. **74,250** records per class, respectively.

Table 3.6: Class distribution in the modelling dataset (N = 495,000) and 70/30 split.

	Totals		70/30 split	
Class	Count	%	Train (70%)	Test (30%)
Normal	247,500	50.00%	173,250	74,250
Attack	247,500	50.00%	173,250	74,250

As a stress test, we applied SMOTE only to the training split of the modelling dataset; since the modelling set is already 50/50, SMOTE did not alter class totals but perturbed the decision boundary slightly (Chawla et al. 2002). Evaluation used the unmodified test split.

3.4.0.5 Feature ranking (RFE with Decision Tree).

To understand which variables help most, we apply recursive feature elimination (RFE) with a Decision Tree base learner on the integrated dataset (process + network) (Guyon et al. 2002). In our runs, process variables (e.g. tank level and flow sensors) ranked higher than packet-only fields for the intrusion-detection task on SWaT. We therefore use process data as PI during training to teach a network-only detector at deployment.

3.4.0.6 Label alignment

Two practical label issues arise in SWaT. First, some process records are marked as "attack" even when the physical behaviour does not yet (or no longer) reflect an attack—i.e., the label boundary leads or lags the observable change (also noted in prior work) (Bernieri et al. 2019). Second, network labels appear to be inherited from process timestamps, introducing small timing mismatches between streams.

To avoid relabelling the dataset while reducing edge effects, we use a conservative boundary filter: for each labelled attack window, we discard a short guard band at the beginning and end of the window when training and evaluating. This removes the slices where misalignment is most likely, without altering the original labels. Figures 3.2 and 3.3 illustrate typical cases: in Attack 36, the level sensor (LIT101) begins changing slightly before the indicated start, and returns to baseline before the indicated end; in Attack 1, the actuator (MV101) flips state earlier than the annotated boundary. These patterns match known "stealthy" behaviours in ICS, where small command sequences can create process impact with subtle network traces(Cárdenas et al. 2011).

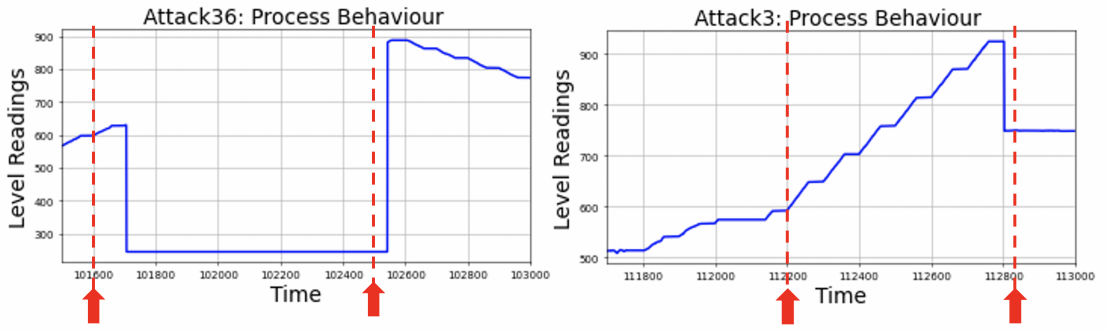

Figure 3.2: Misalignment between indicated attack endpoints (red) and process change (blue) for Attacks 3 and 36 in SWaT.

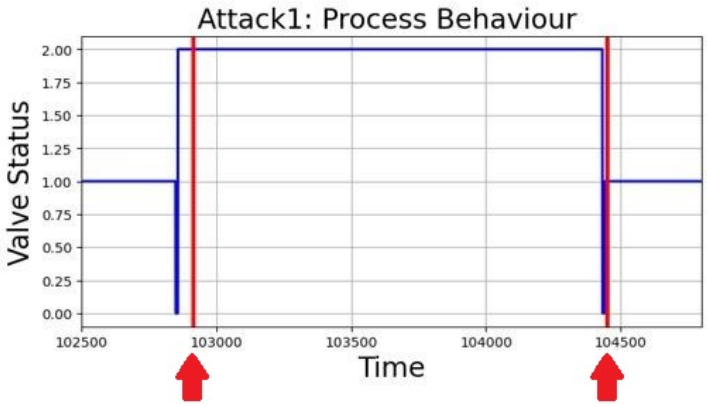

Figure 3.3: Misalignment between indicated endpoints (red) and process change (blue) for Attack 1 in SWaT.

3.5 Evaluation metrics

We used four evaluation measures to report the predictive performance of the proposed intrusion detection methods. This includes:

- **Classification accuracy**, measures all of the correctly identified cases,

$$\frac{TP + TN}{TP + FP + TN + FN,} \tag{3.1}$$

  where TP, FP, FN and TN denote true positives, false positives, false negatives and true negatives, respectively

- **Precision**, the ratio of correctly predicted positive records to the total predicted positive records,

$$\frac{TP}{TP + FP} \tag{3.2}$$

- **Recall**, the ratio of correctly predicted positive records to all data records in a class,

$$\frac{TP}{TP + FN} \tag{3.3}$$

- **The F1-score**, it conveys the balance between precision and recall.

$$\frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \tag{3.4}$$

3.6 Machine Learning Algorithms Used

3.6.1 Learning Using Privileged Information

Supervised classical ML algorithms aim to learn the distribution pattern of labelled training data presented in n number of training pairs (x_i, y_i), where

$i = 1,...,n$, $x_i \in X$, and $y_i \in \{+1,-1\}$. During training, a mapping function $f : X \rightarrow +1,-1$ is formulated that can map an input instance (x_i) to a predicted output (y_i) with the lowest error possible. In classical supervised learning problems, the same data features are used for both training and testing (runtime). In some pattern recognition problems, there may be additional helpful information about the training samples that will not be available during the testing phase. Such data would often be discarded by classical ML algorithms since models have been trained based on training input features only. Recently, there has been a trend in designing ML models that incorporate this additional information (referred to as "privileged information"), alongside the main the training samples. The framework of Learning Using Privileged Information (LUPI) was originally proposed by Vapnik and Vashist (Vapnik and Vashist 2009b; Vapnik 2006) in the context of the Support Vector Machine (SVM) classifier, where a triplet of training data is provided $(x_i, x_i*, y_i), i = 1,...,n$, $x_i \in X$, $x_i^* \in X^*$, and $y_i \in \{+1,-1\}$. Similar to classical ML, the goal is to find a function $f : X \rightarrow +1,-1$ that can predict labels with the lowest error possible. The idea is that the privileged information might improve the learning process and help the ML model converge to a better decision boundary in the input space.

3.6.2 Knowledge Transfer

SVM (Cortes et al. 1995), is a popular supervised learning algorithm for solving non-linear classification problems. The aim is construct a non-linear hyperplane with maximum margin that separates two classes (in the case of binary classification). The SVM allows the decision margin to make some violations known as slack variables (ξ_i). The task here is to find a decision function $f(x) = sgn[\langle w, x\rangle + b]$, where $w \in X$, $b \in \mathbb{R}$ and they are obtained by solving the

following optimization problem

$$\min_{w,b,\xi_i} \frac{1}{2}\|w\|_2^2 + \gamma \sum_{i=1}^{n} \xi_i \tag{3.5}$$

under the constraints,

$$\forall \quad 1 \leq i \leq n, \ [y_i\langle w, x_i\rangle + b] \geq 1 - \xi_i, \quad \xi_i \geq 0, \tag{3.6}$$

where $\gamma \geq 0$ is a hyper-parameter that controls the trade-off between margin maximization and margin violation. If the slacks ξ_i are all equal to zero then we call the set of given examples separable, otherwise they are non-separable. In SVM+ (Vapnik and Vashist 2009b; Vapnik 2006; Fouad 2013), which is based on LUPI, the additional information $x_i^* \in X^*$ will be available during training but not at the test stage. Unlike SVM which uses a correcting slack variable ξ_i, the SVM+ uses a slack function $\xi_i = [\langle w^*, x_i^*\rangle + b^*]$, where $w^* \in X^*$, $b^* \in \mathbb{R}$ and they are obtained by solving the following optimization problem:

$$\min_{w,w^*} \frac{1}{2}\|w\|_2^2 + \frac{\rho}{2}\|w^*\|_2^2 + \gamma \sum_{i=1}^{n} [\langle w^*, x_i^*\rangle + b^*] \tag{3.7}$$

under the constraints,

$$\begin{aligned} \forall \quad 1 \leq i \leq n, \quad [y_i\langle w, x_i\rangle + b] \geq 1 - [y_i\langle w^*, x_i^*\rangle + b^*], \\ [\langle w^*, x_i^*\rangle + b^*] \geq 0 \end{aligned} \tag{3.8}$$

In SVM+, correcting functions control the slack variables based on the privileged information. The objective function of SVM+ contains two hyper-parameters $\gamma, \rho > 0$. The ρ is a non-negative parameter that reflects the imposition of smoothness in the slack model.

3.6.3 Margin Transfer

Margin Transfer SVM is another implementation of Learning Using Privileged Information (LUPI) proposed in (Sharmanska et al. 2013). It focused on transferring knowledge about the learning problem to improve the predictive solution. Similar to SVM+, the aim of Margin Transfer SVM is to distinguish the easy and hard examples. For a Classification problem, Sharmanska et al. (ibid.) suggested to train an ordinary SVM using privileged information X^* and use the achieved prediction function $f^*(\mathbf{x}^*) = \langle \mathbf{w}^*, \mathbf{x}^* \rangle$ to calculate the margin distance of the training samples to the classifying hyperplane in the same space $\rho_i := y_i f^*\left(\mathbf{x}_i^*\right)$. This computed margin can be transferred to an ordinary SVM on Original space instead of exploiting a constant margin of 1. for solving the optimisation problem in the standard space.

$$\underset{\mathbf{w}\in\mathbb{R}^d, \xi_i \in \mathbb{R}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i$$

subject to, for all $i = 1, \ldots, N$

$$y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq \rho_i - \xi_i \quad \text{and} \quad \xi_i \geq 0$$

In this approach not only the performance of Margin Transfer SVM has been validated in the privileged space and enhance the training of a classifier in the original space, it also can identify the easy and hard examples both in privileged space and original space. In contract, SVM+ differentiate the easy and difficult examples by learning the the slack function within privileged space. As stated in $\xi_i = \left\langle \mathbf{w}^*, \mathbf{x}_i^* \right\rangle + b^*$ the label information is not involved in formulating slack function, therefore, SVM+ performance is not validated in privileged space.

3.6.4 Distillation

The idea of distilling knowledge has been proposed By(Hinton et al. 2015). It has been developed based on the fact that almost any machine learning algorithm's performance can be improved in an ensemble structure by training many different models on the data and then averaging their prediction, or by using larger datasets. However, such improvement comes with computational cost and complexity. The distillation technique proposes a methodology to extract and compress class knowledge (hard label) and the probability vector of each class (soft label) to a smaller model. Distillation is used to transfer knowledge from a complex Neural Network model to a smaller one while saving accuracy while reducing the computational cost considerably. For an N-class classification task with data as

$$\{(x_i, y_i)\}_{i=1}^{n}, x_i \in \mathbb{R}^d, y_i \in \Delta^c$$

where Δ^c is an N-dimensional probability vector, the aim would be to learn the following function in which $\mathcal{F}_t$ is a function from $\mathbb{R}^d$ to $\mathbb{R}^c$.

$$f_t = \arg\min_{f \in \mathcal{F}_t} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, \sigma\left(f\left(x_i\right)\right)\right) + \Omega(\|f\|) \tag{3.9}$$

Here σ is a *Softmax* function that operates from $\mathbb{R}^c$ to Δ^c and Ω is an increasing function that works on regularization.

$$\sigma(z)_k = \frac{e^{z_k}}{\sum_{j=1}^{c} e^{z_j}} \tag{3.10}$$

Cross entropy loss will be measured by function ℓ for all $1 \leq k \leq c$:

$$\ell(y, \hat{y}) = -\sum_{k=1}^{c} y_k \log \hat{y}_k \tag{3.11}$$

In order to distill the previously learnt f_t in to f_s, (Hinton et al. 2015) proposed following optimization problem to be solved:

$$f_s = \underset{f \in \mathcal{F}_s}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} [(1-\lambda)\ell(y_i, \sigma(f(x_i))) + \lambda \ell(s_i, \sigma(f(x_i)))] \tag{3.12}$$

where s_i is the soft labels obtained from f_t on the training data

$$s_i = \sigma(f_t(x_i)/T) \in \Delta^c \tag{3.13}$$

The temperature parameter $T > 0$, control the softness in the prediction of the class probability from f_t. The imitation parameter $\lambda \in [0, 1]$ balance the importance of the soft predicted label s_i and the true hard predicted label y_i. Suggested in Eq.3.13. increasing the temperature will result in predicting softer class probability which will identify label dependencies. By using the proposed distillation method, $f_s \in F_s$ will be used that is simpler than f_t therefore, resulting in faster prediction during run-time. (Lopez-Paz et al. 2015) merged Hinton's distillation technique with Vapnik's Privileged information and proposed generalised distillation. With a data such as $\left\{\left(x_i, x_i^\star, y_i\right)\right\}_{i=1}^{n}$, the knowledge is extracted by a teacher model learned $f_t \in F_t$ on privileged information that is a pair of $\left\{\left(x_i^\star, y_i\right)\right\}_{i=1}^{n}$ using Eq.3.9. This knowledge is then compressed through computing teacher soft labels using Eq.3.13. for $x_i^\star$ and $T > 0$. To pass this to the student model, the student should learn $f_s \in F_s$ on the input-output pairs of $\{(x_i, y_i)\}_{i=1}^{n}$ and $\{(x_i, s_i)\}_{i=1}^{n}$ using Eq.3.12. and $\lambda \in [0, 1]$. This approach decreases the computational complexity by following Hinton's distillation method and

improves the accuracy of the classification over the standard dataset by incorporating Vapnik's privileged information. Generalised distillation can be reduced to Hinton's distillation if $x_i^\star = x_i$ for all $1 \leq i \leq n$.

3.6.5 Transfer Learning

Transfer learning is developed based on the fact that humans use their learnt knowledge from one task or problem to solve other problems (Pan et al. 2009). Such motivation suggests retraining a previously learnt model for solving a new task. Classical ML algorithms use labelled or unlabeled data for learning a specific task from scratch and aim to predict future data for that specified task during run-time (Yin et al. 2006; Baralis et al. 2007). However, transfer learning is a technique that transfers knowledge from a learnt task to improve generalization in a target task especially when the available data for the target task does not provide enough information for developing a high-quality model.

Suppose a classification task in domain $\mathcal{D}$, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, that includes a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$. Training data that consisting of pairs $\{x_i, y_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ would be used to learn the predictive function $f(\cdot)$ such that $f(x)$ predicts the corresponding label of an unseen instance x. This function, $f(x)$ can be presented as $P(y \mid x)$. Suppose there is a source domain $\mathcal{D}_S$, and its data is denoted as $D_S = \left\{\left(x_{S_1}, y_{S_1}\right), \ldots, \left(x_{S_{n_s}}, y_{S_{n_s}}\right)\right\}$, where $x_{S_1} \in \mathcal{X}_S$ is the feature instance and $y_{S_i} \in \mathcal{Y}_S$ is the corresponding class label. Similarly consider a target domain, D_T, data of each is denoted as $D_T = \left\{\left(x_{T_1}, y_{T_1}\right), \ldots, \left(x_{T_{n_T}}, y_{T_{n_T}}\right)\right\}$, where the input x_{T_1} is in $\mathcal{X}_T$ and $y_{T_i} \in \mathcal{Y}_T$ is the corresponding output.

One definition for Transfer Learning presented in the work of (Pan et al. 2009) suggests that for a source domain $\mathcal{D}_S$ and corresponding learning task $\mathcal{T}_S$, as well as a target domain $\mathcal{D}_T$ and corresponding learning task $\mathcal{T}_T$, transfer learn-

ing's objective is to utilize the knowledge available in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$ and improve the learning performance of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$.
Two approaches for implementing transfer learning were suggested by (Goodfellow et al. 2017) as follow:

- Weight Initialization This method relies on the information derived from the first task data for training a model on the target task. In this approach, the weights from the learnt model will be used as the initial weights for training the target model therefore, in this process transfer learning can be considered as a weight initialization strategy.

- Feature Extraction When the weights from the learnt model cannot be adapted to the target task, transfer learning can be used as a feature extraction strategy. In this approach, new layers after previously trained layers of a neural network will retrain with target data.

In general, when features of one task correspond to the fundamental factors in another task, transfer learning can be of use to reduce training time that leads to a likely faster or a superior solution due to lower generalisation error.

3.7 Summary

This chapter details the evaluation of a proposed framework designed to enhance the performance of NIDS within ICS environments. The evaluation leverages the Secure Water Treatment (SWaT) test-bed, a scaled-down water treatment plant, and its associated datasets to assess the framework's efficacy.
First SWaT test-bed is introduced, outlining its hierarchical architecture and operational capabilities. The specific datasets used in the evaluation is then

discussed, highlighting the inclusion of both network and process data collected during normal operations and simulated attack scenarios.

A taxonomy of attack types present in the SWaT dataset is provided, categorizing attacks based on their target location and impact on the physical process. The rationale behind selecting specific attack scenarios for evaluating the framework is also explained, emphasizing the focus on attacks that aim to disrupt process control.

This chapter then delves into the data pre-processing steps undertaken to prepare the datasets for analysis. These steps include data normalization, handling of categorical variables and addressing class imbalance as well as the use of process data as "privileged information" during the training phase of the machine learning models.

Furthermore, the chapter identifies and discusses inconsistencies discovered in the labelling of attack data within the SWaT dataset. These inconsistencies, related to both process and network data labelling, are addressed through a filtering approach to ensure the accuracy of the evaluation results.

Finally, the chapter provides a comprehensive overview of the machine learning algorithms employed in the study. Techniques such as Learning Using Privileged Information (LUPI), Knowledge Transfer, Margin Transfer, Distillation, and Transfer Learning are discussed, outlining their theoretical foundations and practical applications in the context of intrusion detection.

In summary, this chapter establishes the experimental setup for evaluating the proposed NIDS framework, encompassing the test-bed, datasets, attack scenarios, data pre-processing techniques, and machine learning algorithms employed. It ensures a robust and reliable evaluation of the framework's ability to enhance intrusion detection capabilities in ICS environments.

Chapter 4

A framework for ML-based network intrusion detection in ICS

4.1 Generic Overview of an ICS Network Structure

The main difference between ICS and traditional information systems is the close relationship with the physical world. We refer to the ICS architecture described in chapter 2 (Figure 2.1). In this chapter, we collect network traffic at L2 and historian data at L3, which is where PINID learns correlations.This chapter motivates and specifies the Process-Informed NIDS (PINID). We focus on attacks that impair process control, especially Brute Force I/O, (T0806), (MITRE Corporation 2025a) and Unauthorized Command Message, (T0855), (MITRE Corporation 2025n), because they alter the physical process with subtle network traces. Figure 4.1 presents the landscape of MITRE ATT&CK for ICS and highlights where PINID aims to help.

Figure 4.1: Adversarial Tactical Goals in Compliance with MITRE ATT&CK Matrix for ICS Protection(Alexander et al. 2020). This diagram draws attention to the focal point of this research, with particular emphasis on the techniques employed when aiming at Impair Process Control strategies, Brute Force IO and Unauthorised Command Messages, and recognise pertinent data sources for exploring such attacks.

4.2 ICS Attack Surface

According to a study conducted by researchers and professionals in the cyber security field, 12 adversarial tactics have been identified for attacking ICS (Alexander et al. 2020). Today an active area of research in the cybersecurity domain involves the integration of ML into the field of ICS for intrusion detection and anomaly detection purposes. Researchers are actively exploring and experimenting with different ML techniques to improve the security of ICS. Their proposed methods target diverse tactics that attackers use to reach their desired goal. These adversarial tactics and the incentive of the adversaries are:

1. Initial Access (TA0108): The adversary's primary goal is to infiltrate the ICS environment, aiming for initial access (MITRE Corporation 2025j).

2. Execution (TA0104): The adversary's goal during execution is to run code or manipulate system functions, parameters, and data without authorisation (MITRE Corporation 2025f).

3. Persistence (TA0110): The adversary's primary objective is to persistently maintain their presence in the ICS environment, in order to continue their malicious activities (MITRE Corporation 2025l).

4. Privilege escalation (TA0111): The adversary's objective includes gaining higher-level permissions, which is known as privilege escalation (MITRE Corporation 2025m).

5. Evasion (TA0103): The adversary is employing tactics to bypass and elude the existing security defences (MITRE Corporation 2025e).

6. Discovery (TA0102): The adversary is actively searching for information in order to assess and identify potential targets within the industrial network (MITRE Corporation 2025d).

7. Lateral Movement (TA0109): The adversary is actively attempting to navigate and infiltrate the ICS environment (MITRE Corporation 2025k).

8. Collection (TA0100): The adversary's primary objective is to obtain valuable data and acquire comprehensive domain knowledge related to the ICS environment, which will serve as crucial information to advance their goal (MITRE Corporation 2025b).

9. Command and Control (TA0101): The adversary's objective is to establish communication with compromised systems, controllers, and platforms that have access to the ICS environment, with the intention of gaining control over them (MITRE Corporation 2025c).

10. Inhibit Response Function (TA0107): The main objective of the adversary is to hinder the proper functioning of the safety, protection, quality assurance, and operator intervention functions in responding effectively to any failure, hazard, or unsafe state (MITRE Corporation 2025i).

11. Impair Process Control (TA0106): The adversary is attempting to engage in activities that involve manipulating, disabling, or damaging physical control processes (MITRE Corporation 2025h).

12. Impact (TA0105): The adversary's goal is to manipulate, interrupt, or even go as far as destroying the ICS systems, the valuable data they hold, and the environment in which they operate (MITRE Corporation 2025g).

By thoroughly studying literature on the detection of ICS cyberattacks and analysing notable and recent cyberattacks on ICS systems,such as the Stuxnet cyber attack on Iran's nuclear facilities in 2010 (Langner 2011), the cyber attack on the Ukrainian power grid in 2015 (Case 2016), and the recent cyber attack

on the Colonial Pipeline in the USA(Hobbs 2021), we have identified a pattern wherein attackers consistently opt for Impair Process Control techniques. Impairing Process Control refers to the various techniques used by malicious actors to disrupt the control logic of systems and cause detrimental outcomes for the processes being controlled within the targeted environment. The potential targets for manipulation could include procedures or parameters that have an active impact on the physical environment. These techniques are not limited to prevention or manipulation of reporting elements and control logic, they can also involve various other approaches. When an adversary modifies the functionality of a process, it is important to note that they might obfuscate the results as well, and these outcomes often provide clear evidence of their influence on a product or the surrounding environment. The safety of operators and downstream users is at risk because of the direct physical control exerted by these techniques, and this can lead to the activation of response mechanisms, such as a safety shutdown. To successfully control processes and cause impact, adversaries utilise techniques to inhibit such mechanisms. This research is exploring the efficacy of supervised ML techniques in augmenting the detection of cyber-attacks where attackers attempt to manipulate, interrupt, or destroy ICS (Impair Process Control). We prioritise two Impair Process Control techniques: Brute Force I/O (T0806)—repetitively changing I/O point values to manipulate a process—and Unauthorized Command Message (T0855)—issuing commands outside intended bounds. Both can produce faint network signals yet meaningful process effects.

4.2.1 Brute Force IO

In the enterprise network, the tactic known as brute force is employed to gain unauthorized access to a system. This involves repeatedly guessing login cre-

dentials until the correct combination is identified. This tactic is often used in conjunction with other attack methods, such as phishing or social engineering, to obtain login credentials or other sensitive information. In Industrial Control Systems (ICS), the tactics employed by adversaries involves repeatedly or sequentially altering I/O point values, thus allowing them to execute specific actions. By repetitively altering either a range of I/O point values or a single point value, brute force I/O is used to manipulate a process function. The approach chosen will vary depending on the objectives of the adversary and their level of knowledge about the target environment. When attempting to brute force a range of point values, the adversary may inadvertently cause an impact, without having a specific target in mind. On the other hand, if the focus is solely on one point, it is possible for the adversary to cause instability in the corresponding process function. The reason brute force IO attacks can be so harmful in ICS environments is that they give attackers the ability to gain control of crucial systems, which can then result in physical damage or significant disruption to operations. As an illustration, let us consider a scenario where an attacker successfully infiltrates the control system of a power plant or water treatment facility. In such a situation, the potential consequences could range from triggering a widespread blackout to deliberately contaminating the water supply. In order to enhance security against brute force IO attacks in ICS environments, organizations should consider implementing robust password policies and incorporating multi-factor authentication. These measures can significantly increase the difficulty for attackers to guess or obtain login credentials. Furthermore, organizations need to stay vigilant by consistently monitoring their systems for any potential signs of suspicious activity. This involves being alerted to repeated login attempts coming from the same IP address and being mindful of any unusual login patterns that might indicate a security breach. In ICS

environments, there are several security measures that can be implemented to prevent brute force IO attacks. These include network segmentation, which divides the network into smaller sections, as well as the use of intrusion detection and prevention systems. Regular security assessments and audits should also be conducted to ensure the effectiveness of these measures. Implementing a proactive security strategy empowers organizations to minimize the risk of successful brute force IO attacks, thereby safeguarding their critical systems and operations.

4.2.2 Unauthorised Command Message

In ICS, the unauthorised command message tactic is employed by attackers who try to send or execute commands that are not allowed within the ICS environment. The main objective of this tactic is to manipulate or disrupt the control system's normal operation. Attackers may try to send unauthorised commands to gain control over devices like programmable logic controllers (PLCs) or remote terminal units (RTUs). The intention behind these commands is to change process parameters, adjust control logic, or potentially disrupt the functioning of critical infrastructure. Attackers could exploit this technique through range of methods, such as taking advantage of vulnerabilities present in the ICS network, compromising user accounts that have higher privileges, or employing social engineering tactics to gain control over the control system. The consequences of unauthorised command messages in ICS can be severe, ranging from operational disruptions to safety hazards and even potential physical damage. To illustrate the consequences, imagine an attacker gaining unauthorised access and using it to send commands that manipulate the flow of a pipeline, alter temperature settings in a chemical plant, or disrupt power distri-

bution in a grid. In order to reduce the risk of unauthorised command message attacks in ICS, it is recommended to implement effective security measures. One of the key measures to ensure security is network segmentation, which involves isolating critical control systems. Besides that, implementing strong access controls that include multi-factor authentication can enhance the security measures. In order to identify any potential vulnerabilities, regular security assessments and audits should be conducted. A system of continuous monitoring would enable identifying and investigating any suspicious activities or anomalies within the ICS environment. In addition, implementing user awareness training and establishing strict change management processes can be effective measures to prevent the execution of unauthorised commands within the ICS infrastructure. A proactive measure in safeguarding the industrial network , is to implement intrusion detection and prevention systems, which detect and prevent unauthorised access or malicious activities.

In the event of a Brute Force attack and Unauthorised Command Message and alteration of the system, an irregular pattern will be created in the usual flow of network traffic and data held in the Historian. These surprising coincidences that are detectable in both databases concurrently could be seen as an indicator of potential breach. Thus, this can be employed to recognise and uncover both of these classifications of attack on an ICS system.

4.3 Process Informed Network Intrusion Detection Framework (PINID)

In order for external attackers to manipulate the process, they must gain access to the organisation through an enterprise network. By utilising their understanding of the organisational infrastructure, industrial devices and equip-

ment, communication systems, as well as the employees and staff, they can detect weaknesses and vulnerabilities within the organisation. By leveraging this information, they would possess the capability to direct their efforts towards a systems or users within the enterprise network, gaining illicit access to their system. This can be achieved through the execution of targeted attacks, such as phishing attacks, ultimately resulting in the establishment of a command-and-control connection that remains undetected. Their ability to remain silent within the network enables them to gather additional information on systems and user credentials. This information can then escalate privileges and gain access to critical systems leaving no significant trace. By exploiting identified vulnerabilities in critical systems, they can effectively disturb the process and manipulating operators' HMI view. The current methods for detecting adversarial attempts to impair process control tactics are limited to process BAD and Network Intrusion Detection (NID).

The application of an anomaly detection approach in the ICS environment, with a specific emphasis on process behaviour, has been proposed by a team of researchers. The suggested strategies encompass a combination of supervised and unsupervised machine learning techniques, including one-class, binary, and multi-class categorization (Maglaras and J. Jiang 2014; Suaboot et al. 2020) as well as time-series prediction (D. Li et al. 2019). The activation of an alert will be triggered by any deviation from the expected behaviour in these methods. However, given the potential for cyber incidents to alter system behaviour, besides system failures or equipment malfunctions, a comprehensive investigation is needed to determine the origin of each detected anomaly and obtain an accurate assessment of the security state. It is important to mention that the technique of focusing on Process behaviour presents a greater challenge in detecting cyber-attacks during the initial stages of reconnaissance, as the attacker

is still in the process of gathering information and has not yet caused significant changes to the process behaviour. Various research groups have explored the analysis of ICS network traffic to detect network intrusion, operating under the assumption that any external cyber incident would eventually penetrate the ICS environment through the network. The researchers utilised parallel Machine Learning techniques, incorporating the supervised approaches noted by Valdes et al. (2009) and the unsupervised methods as described by Almalawi, Yu et al. (2014). The drawback of this approach lies in its ability to detect internal attacks and those that can bypass established security measures without affecting the network's status, such as stealthy attacks. As stated by Cárdenas et al. (2011), attacks of this nature have been found to have limited influence on the manipulation of sensor data, persisting over an extended period and posing a risk of significant damage. If this minor modification is introduced into the control loop, it has the potential to manifest in the behaviour of the process. Given that the network status remains unaltered, the immediate visibility of the impact on network traffic may be limited. By categorising all device malfunctions or failures as behaviour anomalies, the first method is likely to detect cyber incidents. However, the second technique is incapable of identifying attacks that have successfully eluded security countermeasures, even if their effect on the system is negligible. The design of our proposed framework, illustrated in Figure 4.2, is aimed at enhancing the detection capabilities of NIDS specifically against attacks targeting ICS Process Control. This framework is structured to align with the hierarchical network of ICS networks, taking into account the distribution of components across different levels as defined by the Purdue model Williams 1994. For our purposes, we focus on levels 0-3 of the Purdue model, encompassing ICS components, while excluding Enterprise levels dedicated to IT and business functions.

PINID follows LUPI paradigm: during training, we expose the model to process data aligned to network features; at runtime, the model uses network features only. LUPI has been shown to improve generalisation when rich valuable data are available at training time (S. Yan et al. 2023; Z. Gao et al. 2024; S. Yang et al. 2022; X. Li et al. 2020).

Data collection for the development of our supervised Machine Learning (ML)-based NIDS is conducted at level 2, the supervisory level. This level provides access to network traffic exchanged between servers and control systems, as well as operator stations situated at level 3. Simultaneously, process data is acquired from the historian server at level 3. Despite simultaneous data collection, variations in data capturing frequencies from the network and storage rates within the historian server, as elaborated in section 1.2, necessitate a label alignment process. This process ensures accurate correlation between network traffic and corresponding process data based on attack occurrence time, facilitating effective manual labelling. The aligned datasets are then integrated into a unified database using common entities as linking points. Both network and process features from this integrated dataset serve as inputs for training the ML-based NIDS. However, it's important to note that during the validation phase and real-time operation, the trained model relies solely on network features for attack detection. This approach ensures the practicality and efficiency of the NIDS in real-world deployments. As illustrated in Figure 4.3, this framework comprises four distinct stages.

The first phase, known as **Collection and Analysis** , starts by gathering network traffic and process data from the industrial setup (1.1). Network traffic is to be obtained from the supervisory network, while the process data should be acquired from the historian server. To ensure the accuracy of the labels, a thorough analysis of these datasets is necessary, focusing specifically on detecting

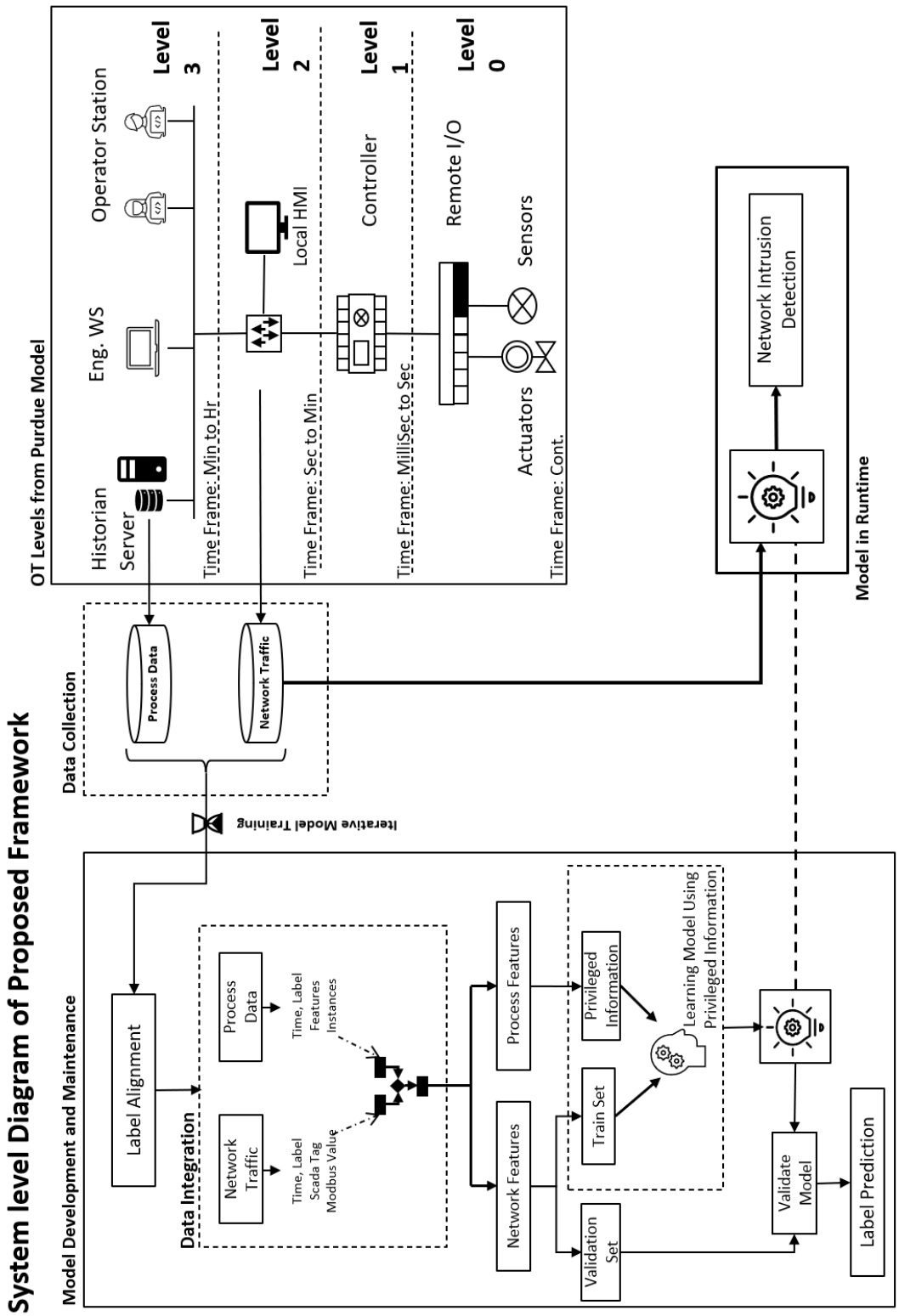

Figure 4.2: System level Diagram of the Process-Informed Network Intrusion Detection Framework (PINID) for the SWaT Testbed Using Process Data as Privileged Information.

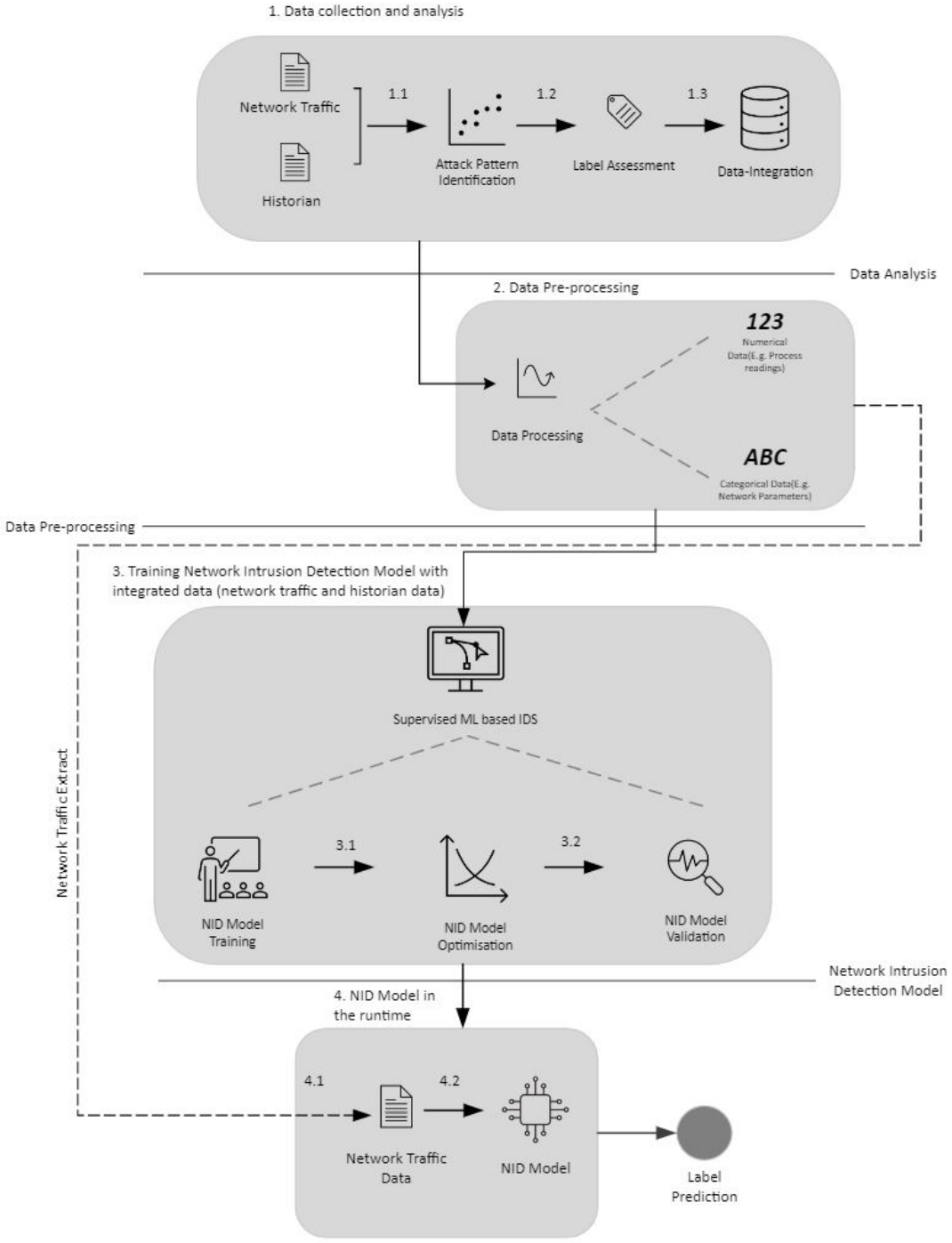

Figure 4.3: PINID Process Flow Diagram

any anomalous patterns(1.2). The patterns that have been identified should be compared with both the attack specification and the normal state. Given that this framework is focused on supervised network intrusion detection, the accuracy of the labels holds significant importance and has the potential to impact the predicted outcome. It is imperative that the verification of dataset labelling is performed by a process specialist who has a thorough understanding of the specific process for which the NID is being designed (1.3). Integrating the two datasets should occur by considering the timestamp from each dataset(1.4). The next stage encompasses the **pre-processing of data** to prepare it for the training and testing of a machine learning algorithm. The conversion of categorical data to numerical format is essential for network traffic data. In addition, the normalisation of all numerical data ensures that the machine learning algorithm's prediction is influenced equally by each feature. The imbalanced distribution of the data classes must be addressed at this stage through implementing data sampling techniques. Finally, the data should be divided into two sections, namely training and validation (70%), and testing (30%). In the third stage, the **training of the supervised machine learning based intrusion detection system (IDS) model** will take place. To proceed with this phase of the process, it will be necessary to utilise the testing and validation segment of the integrated data (which includes network and process features). This segment will be divided into a training set (70%) to train the algorithm and a validation set (30%) to validate the optimised model. To attain the highest level of performance for the model, it is imperative to fine-tune the algorithm parameters through an optimisation process (3.2). The ML based NID model will be fully operational in the test environment after the successful completion of this stage. The fourth and ultimate phase of this framework **NID model in the runtime** involves the model's execution in the test environment. Given that this model is specifically

designed for network intrusion detection, only the network features will be received during runtime. Consequently, only the network features from the test partition of the data prepared during stage 2 of the framework will be utilised by ML based NID for attack detection. It is important to note that, during the training phase, the model was trained on both network and process features.

4.4 Summary

This chapter provides an analysis of ICS security, focusing on the limitations of existing anomaly detection methods and proposing a novel framework for enhanced intrusion detection. The chapter begins by differentiating ICS from traditional information systems, highlighting the direct link between ICS and physical processes and the resulting security implications as well as the hierarchical architecture of ICS, which although efficient, it extend the ICS attack surface.// This chapter then delves into the vulnerabilities of ICS, outlining twelve adversarial tactics and emphasizing the prevalence of Impair Process Control techniques. Brute Force IO and Unauthorized Command Message attacks are explained in detail, illustrating their mechanisms and potential consequences. Existing security measures and their limitations are also discussed.// The chapter analyses current ICS security research, particularly Process BAD and NIDS. BAD's reliance on detecting significant deviations from normal behaviour is deemed insufficient for identifying subtle manipulations, while NIDS focus on network traffic analysis is seen as susceptible to stealthy attacks. To address these limitations, we propose PINID framework. PINID integrates network and process data during the training phase of its machine learning-based intrusion detection system, enabling it to discern subtle correlations between network activities and process behaviour. This approach enhances PINID's

ability to detect attacks that might evade traditional methods.// The chapter concludes by outlining the training process of PINID's supervised machine learning-based IDS model and emphasizing the framework's unique use of process data as "privileged information" during training. This approach enables PINID to develop a more comprehensive understanding of the relationship between network traffic and process behaviour, bolstering its accuracy and efficiency in detecting real-world attacks.

Chapter 5

Result and analysis

5.1 Experiments and Findings

This research has proposed a new framework for ML-Based NIDS in ICS. In distinction to other proposed methodologies in this research domain, which primarily focus on either network traffic data or process data in an ICS environment, the objective of this research was to take into account the principal characteristics of ICS in controlling a physical process through a highly connected network architecture. Therefore, the two data types (network traffic and process data) were employed in the training process for the development of a ML-based NIDS model. However, to conform to the expected operational definition of a NIDS, during the runtime, the model is solely exposed to the network traffic . The core structure of the proposed framework is built upon the Machine Learning concept known as Learning Using Privilege Information (LUPI). Validation of the framework was conducted by applying various techniques that were specifically adjusted for LUPI. This chapter will begin with a brief summary of the testbed and datasets employed in this research (for a detailed explanation, please refer to section 5.1.1). Following that, it will present the

design, implementation, and analysis of the results of each experiment. Lastly, the chapter will be brought to a close with a concise summary that will present the key findings.

5.1.1 Brief summary of the Testbed

The SWaT test bed, is a small water treatment plant with six process stages, delivering 5gal/min of double-filtered water. Each stage of the process in the SWaT architecture is managed by a redundant pair of PLCs. The data collected from this testbed consists of network traffic and historian data. SWaT network dataset consists of network traffic captured from SCADA and PLCs interaction. it includes 19 selected features in the dataset for intrusion detection (for the detail list of the features please refer to section 3.2) each logged at a high rate of milliseconds. SWAT Historian dataset includes readings from 51 field instruments recorded every second. The SWaT datasets includes 36 attack scenarios with physical impacts on the process which are categorised into 4 groups based on the location of the attack point in the process stage and the number of compromised elements in each attack as *Single Stage Single Point (SSSP)*, *Single Stage Multi-Point (SSMP)*, *Multi-Stage Single Point (MSSP)* and *Multi-Stage Multi-Point (MSMP)*. This chapter's experiments considered a selection of attacks on actuators and sensors from these 4 categories.

5.1.2 Data subset used for each experiment

As outlined in sections 3.2–3.5, our modelling dataset has **N=495,000** merged network–process pairs with a **50:50** class balance by design (equal per-attack sampling and an equal number from Normal). For the single-attack scenario, the most we can use is the per-attack draw (n = 49,500 Attack + 49,500 Normal

= **99,000** records). For the multi-attack studies, we can use the full set (**495,000** records).

Neural networks usually benefit from more data and train more stably with larger subsets, so we give them the maximum possible in each scenario to steady optimisation and reduce run-to-run variance (Kaplan et al. 2020). By contrast, SVM+ adds constraints and variables for the privileged features, which raises memory and computational cost versus a standard SVM; we therefore use a smaller but balanced subset to keep training feasible without changing how we evaluate (Vapnik and Vashist 2009a; Lapin et al. 2014).

To keep comparisons fair while respecting compute limits, we follow three simple rules:

(i) all subsets are balanced (Attack:Normal = 1:1) and stratified; (ii) **Neural Network** runs use the largest available subset in each scenario (same size across repeats to stabilise training); (iii) **SVM+ (LUPI)** uses a smaller subset due to higher training cost with privileged features.

This explains the different record counts across techniques. The evaluation protocol itself is unchanged (see section 3.5; 70/30 split; network-only at test).

Experiment	Total records	Class ratio
Neural Network (Distillation)	**49,500**	1:1
Neural Network (Transfer Learning)	**49,500**	1:1
Neural Network (Margin Transfer)	**49,500**	1:1
SVM+ (LUPI)	**9,424**	1:1

Table 5.1: Data subsets used in the *single-attack* scenario (example: Attack 36). All subsets are balanced (Attack:Normal = 1:1).

Experiment	Total records	Class ratio
Neural Network (Distillation)	**99,200**	1:1
Neural Network (Transfer Learning)	**99,2000**	1:1
Neural Network (Margin Transfer)	**99,200**	1:1
SVM+ (LUPI)	**9,424**	1:1

Table 5.2: Data subsets used in the *multi-attack* scenario (five attacks). All subsets are balanced (Attack:Normal = 1:1).

5.2 Impact of label alignment on baseline classifiers

this section presents the performance of common classifiers before vs. after applying the label boundary filter described in section 3.4.0.6. The goal is to quantify how small start/end timestamp misalignments (Figures 3.2–3.3) affect detection performance.

In this sanity check, all models use network-only features for both training and testing, keep the same 70/30 split, and follow the metrics in section 3.5. The "post-alignment" step only trims short guard-bands at the start and end of labelled attacks; we do not change any labels. We therefore observed fewer false positives near those boundaries and a lift in precision, with recall changing little if at all.

Classifier	Accuracy	Precision	Recall	F1-score
KNN	65.057%	66.214%	66.717%	63.972%
	(±2.580%)	(±1.723%)	(±0.858%)	(±0.663%)
LR	59.770%	64.594%	61.602%	60.089%
	(±2.666%)	(±1.817%)	(±1.435%)	(±2.305%)
DT	84.215%	61.570%	63.238%	60.888%
	(±1.686%)	(±2.259%)	(±1.990%)	(±2.226%)
MLP	59.847%	60.754%	59.378%	53.219%
	(±3.726%)	(±9.690%)	(±4.821%)	(±3.269%)
CNN	60.327%	66.084%	61.382%	58.033%
	(±4.249%)	(±6.244%)	(±3.637%)	(±5.693%)
SVM	60.690%	66.696%	60.688%	60.062%
	(±1.202%)	(±3.681%)	(±2.284%)	(±3.019%)

Table 5.3: NIDS Performances Using Network Traffic *Before* Applying the Label Alignment Method. The mean values of Accuracy, Precision, Recall and F1-score, along with standard deviations (±) across 5 training/test re-sampling is reported.

Classifier	Accuracy	Precision	Recall	F1-score
KNN	65.779%	65.151%	65.666%	64.492%
	(±2.164%)	(±2.242%)	(±2.826%)	(±2.308%)
LR	60.913%	61.790%	62.610%	61.107%
	(±1.282%)	(±2.020%)	(±2.153%)	(±1.312%)
DT	84.639%	62.269%	61.814%	64.193%
	(±1.963%)	(±2.429%)	(±1.847%)	(±1.501%)
MLP	65.095%	62.548%	66.192%	62.060%
	(±2.324%)	(±7.562%)	(±1.054%)	(±2.969%)
CNN	61.714%	63.828%	61.726%	60.269%
	(±2.470%)	(±2.811%)	(±1.849%)	(±2.473%)
SVM	62.890%	62.698%	63.598%	61.088%
	(±2.186%)	(±2.986%)	(±2.299%)	(±2.721%)

Table 5.4: NIDS Performances Using Network Traffic *After* Applying the Label Alignment Method. The mean values of Accuracy, Precision, Recall and F1-score, along with standard deviations (±) across 5 training/test re-sampling is reported.

Subsequent sections report single- and multi-attack results under the same evaluation protocol; where privileged information is used (e.g. SVM+), it is during the train-time only, with network-only features at test.

5.3 Single attack scenario results

We evaluate models on a single attack (36) to isolate per-scenario behaviour under TA0106 (Impair Process Control). We use the balanced data subset (Attack:Normal = 1:1) exact record counts for each experiment is listed in Table 5.1.

5.3.1 Knowledge Transfer (SVM+)

This experiment follows the protocol in Section 5.1.2 an Table 5.1 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.

In this experiment, we applied supervised ML models to identify one type of attack (Attack 36) using network data of the SWaT testbed. We particularly evaluated the performance of six classical and popular ML models including, K-Nearest Neighbour (K-NN), Logistic Regression (LR), Decision Tree (DT), Multilayer Perceptron (MLP), one-dimension Convolutional Neural Network (CNN), and Support Vector Machine (SVM) (Géron 2019). (Hyper-)parameters of all classification algorithms were tuned via 5-fold cross-validation on the training set. we then trained the SVM+ (LUPI) model, discussed in section 3.6.2, to identify *Attack 36* using the integrated training set. this model was tested using the network data only. For fair comparisons, the experiments were run five times for each classifier and we reported the average results over five runs. All experiments were run using Python (scikit-learn libraries) and Jupyter hosted on Google's Colab platform. SVM+ trains with network features as standard inputs and aligned historian records as privileged inputs; at test time, it uses network data only. The evaluation protocol, metrics, and subset policy follow section 5.1.2 and section 3.5; label alignment follows section 3.4.0.6.

5.3.1.1 Findings and Analysis

Results obtained from SVM+ algorithm was compared against the six classical ML models, which were trained and tested on network traffic only, without considering the impact of the process data. As reported in table 5.5, on average, the accuracy, precision, Recall and F1-scores results obtained by SVM+ algorithm outperform the results obtained by all classical ML algorithms by

Classifier	Accuracy	Precision	Recall	F1-score
KNN	65.779% (±2.164%)	65.151% (±2.242%)	65.666% (±2.826%)	64.492% (±2.308%)
LR	60.913% (±1.282%)	61.790% (±2.020%)	62.610% (±2.153%)	61.107% (±1.312%)
DT	84.639% (±1.963%)	62.269% (±2.429%)	61.814% (±1.847%)	64.193% (±1.501%)
MLP	65.095% (±2.324%)	62.548% (±7.562%)	66.192% (±1.054%)	62.060% (±2.969%)
CNN	61.714% (±2.470%)	63.828% (±2.811%)	61.726% (±1.849%)	60.269% (±2.473%)
SVM	62.890% (±2.186%)	62.698% (±2.986%)	63.598% (±2.299%)	61.088% (±2.721%)
$SVM+_{LUPI}$	74.2534% (±1.022%)	77.251% (±0.849%)	74.1692% (±1.173%)	73.4782% (±1.375%)

Table 5.5: Single-attack results for SVM+ (LUPI). Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2; metrics section 3.5; label alignment per section 3.4.0.6.

12.49%, 22.57%, 16.71%, and 19.45%, respectively. It is important to highlight that, the accuracy obtained by SVM+ outperforms all baseline ML algorithms except for the DT classifier.

5.3.2 Margin Transfer

This experiment follows the protocol in Section 5.1.2 an Table 5.1 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.

This experiment was completed by teaching a Neural Network model through a gradient descent optimisation process. The Margin Transfer method was employed to alter the learning rate of the examples used for training this NN structure. Two Neural Network models, one with one hidden layer and the other with two hidden layers, were created and trained with the ReLU activa-

tion function. To begin, the NN models were trained using network data, while the learning parameters from the back-propagation of each layer of the network were recorded. The same course of action was repeated with process data as PI and the learning parameters were retained in the same manner. To assess our hypothesis, we have tested the NN models with a data set made up entirely of network data, using only the learning parameters from network data. We measured the performance metrics from the models which had taken learning parameters from PI data against the same experiment dataset. Table 5.6 displays a comparison where *NN* is for a neural network trained with network data and NN_{PI} is a neural network trained with network data with learning parameters obtained from the neural network trained with process data as PI.

Classifier	Accuracy	Precision	Recall	F1-score
$NN(1HL)_{bline}$	47.70% (±0.052%)	88.10% (±0.044%)	42.30% (±0.047%)	57.10% (±0.051%)
$NN(1HL)_{PI}$	80.70% (±0.078%)	83.80% (±0.016%)	94.80% (±0.105%)	88.70% (±0.054%)
$NN(2HL)_{PI}$	82.50% (±0.0003%)	82.50% (±0.003%)	100% (±0.00%)	90.40% (±0.002%)

Table 5.6: Single-attack results for Margin Transfer-Based NIDS. Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2 5.1; metrics section 3.5; label alignment per section 3.4.0.6.

5.3.2.1 Findings and Analysis

During the learning phase, the NN model with Margin Transfer denoted here as NN_{PT}, has been trained using the network data. In addition, this model incorporates the difficulty of the sample information into the learning process. The NN model without Margin Transfer, denoted here as NN_{bline}, has been trained using the network data only without incorporating the difficulty of the sample

information. The detection performances of both NN models were assessed using our evaluation measures and reported in Table 5.6. For fair comparisons, we ran this experiment five times for each NN model and we reported the average results over the 5 runs.

The experimental results show that the $NN(2HL)_{PT}$ method, which incorporates the process data in the learning phase, provides superior performance over the baseline NN model ($NN_{baseline}$) across Accuracy, Recall and F1-score . On average, the results obtained from Margin Transfer method with $NN(1HL)_{PT}$ outperform the baseline model on Accuracy, Recall and F1-score by 42.18%, 57.70% and 35.63% while the $NN(2HL)_{PT}$ outperform the baseline model on Accuracy, Recall and F1-score by 40.89%, 55.38% and 36.84%. Obtained results display the superiority of the Margin Transfer method in PINID framework.

5.3.3 Distillation

This experiment follows the protocol in Section 5.1.2 an Table 5.1 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.
This experiment focuses on examining the effectiveness of the distillation approach for detection of a single attack (explained in section 3.6.4) by distilling the information from privileged process data into a NN model during the learning phase of NIDS.

First a teacher model was trained using the privileged process samples. Then the performance of two similarly structured Deep Neural Network (DNN) models, known as student was studied. These models included two hidden layers and a Softmax layer for each class and were trained by network data. However, the $DNN-Student_D$ model was informed by teacher model's understanding of the process data, while the $DNN-Student_S$ model was trained just

using the network data. The DNN models comprised an input layer with 1000 neurons, besides the two hidden layers with 50 and 10 neurons, respectively, with the ReLU activation function while the Adam optimizer, with a learning rate of 0.001 was used (selected via grid search). Batch size and epoch are respectively selected as 1000 and 10 during optimisation. Since we are using the Softmax for the calculation of the loss function, as discussed in section 3.6.4, we fed the non-normalised probabilities from the last layer of the neural network to the loss function while applying Adam optimizer Kingma et al. 2014. A similar experiment was performed using the CNN model. Like the previous experiment, teacher and student models were structured similarly comprising two 1 Dimensional Convolution layers with the ReLu function, followed by a 1 Dimensional MaxPooling layer, a flattened layer. The standard CNN student model was trained with network data only, denoted here as $CNN - Student_S$. The CNN teacher model was trained with process data, and the learnt pattern was then distilled to the student model $CNN - Student_D$, as trained with network data.
For a fair comparison, we run the training and testing of each of the DNN and CNN models five times and we reported the average testing results over 5 runs in Table 5.7.

5.3.3.1 Findings and Analysis

The result obtained from distilled DNN model and distilled CNN model outperform their baseline models with a considerable improvement in the accuracy, precision, and F1-score while the average results obtained by DNN structured distillation model for accuracy, precision and F1-score outperform the DNN baseline model by 47.24%, 52.60% and 26.39% respectively. CNN structured distillation model outperforms its baseline model for accuracy, precision

Classifier	Accuracy	Precision	Recall	F1-score
$DNN - Student_S$	42.68% (±0.0037%)	46.90% (±0.002%)	72.98% (±0.002%)	57.10% (±0.002%)
$DNN - Student_D$	80.89% (±0.0438%)	98.95% (±0.0102%)	62.37% (±0.0835%)	76.21% (±0.0704%)
$CNN - Student_S$	33.51% (±0.018%)	65.45% (±0.026%)	82.49% (±0.012%)	72.98% (±0.021%)
$CNN - Student_D$	81.76% (±0.026%)	98.68% (±0.008%)	64.40% (±0.026%)	77.77% (±0.042%)

Table 5.7: Single-attack results for Distilation-Based NIDS. Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2 5.1; metrics section 3.5; label alignment per section 3.4.0.6.

and F1-score on average by 59.01%, 33.67% and 6.16% respectively. As displayed in Table 5.7 distilled CNN model outperforms distilled DNN model slightly in the number of True Positive and True Negative predictions of the model and accuracy and F1-score. However, comparing the accuracy, precision and F1-score of the distilled models with their recall score, distillation may not be successful in reducing the number of False Negatives.

5.3.4 Transfer Learning

This experiment follows the protocol in Section 5.1.2 an Table 5.1 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.

The experiment was designed to use the feature extraction capability of transfer learning. To do so a Convolution Neural Network (CNN) model was created including two-Dimensional Convolution layers, a max pooling layer followed by a Deep Neural Network (DNN) with an input layer with 1000 neurons, besides the two hidden layers with 100 and 10 neurons, and the ReLU activation function. The grid search optimisation algorithm was used to was optimise the

model and the filter and kernel size of the convolution layers were set at 5, while an Adam optimizer was used with a 0.001 learning rate. The batch and epoch were set at 1000 and 100 respectively. The CNN model was first trained on the previously selected dataset (comprising both process and network features). After the initial training step, the Convolution layers were frozen and the DNN part of the model was re-trained with network data only. This pre-trained CNN model, denoted here as $CNN_Pretrained$, was then tested on the network data only. The performance of this this model was compared against a similarly structured CNN model that was trained and tested using network data only, denoted here as $CNN_Baseline$. For the fair performance comparison between these two models, the training and testing of each of the pre-trained DNN and DNN models were ran five times and the average testing results over 5 runs were reported in Table 5.8.

Classifier	Accuracy	Precision	Recall	F1-score
$CNN_{Baseline}$	54.934% (±0.068%)	46.412% (±0.293%)	42.415% (±0.126%)	42.415% (±0.126%)
$CNN_{Pretrained}$	50.443% (±0.009%)	35.113% (±0.221%)	50.443% (±0.009%)	34.281% (±0.021%)

Table 5.8: Single-attack results for Transfer Learning-Based NIDS. Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2 5.1; metrics section 3.5; label alignment per section 3.4.0.6.

5.3.4.1 Findings and Analysis

The experimental results show that the Pre-trained CNN and baseline CNN models exhibit almost similar behaviour regarding the accuracy, recall and precision and they were not successful in performing the single attack detection task. However, Pre-trained CNN provides slightly better Recall score performance over the baseline CNN model (trained using network data only).

As presented in Table5.8 although it seems the two models behave almost similarly, the lower standard deviation calculated for the average results of the transferred model suggest that this practice resulted in a slightly more robust model(pre-trained).

5.4 Key findings from Single attack scenario

These experiments presented alternative ML-based NIDS that use the LUPI framework. Unlike classical ML algorithms for anomaly detection which rely upon only one source of data for learning, these ML-based NIDS incorporates process data as privileged information during the training phase. This allows for a more accurate and robust ML-based NIDS than is possible using classic methods, while requiring similar computational resources at run-time because the testing phase only involves one source of data, namely, the network traffic.

5.5 Single attack Scenario : Summary

This chapter presents an empirical evaluation of PINID framework through employing LUPI paradigm to enhance the performance of ML-based NIDS in ICS. The study leverages the SWaT testbed and its datasets, focusing on single attack detection scenarios.

Initially, a feature selection analysis using the DT algorithm highlighted the superior predictive power of process features compared to network features for intrusion detection. This finding justified the use of process data as privileged information during the training phase of LUPI-based NIDS models.

This chapter then presents a series of experiments evaluating different LUPI implementations:

- **Knowledge Transfer**: SVM+ model, trained with both network and process data, outperformed six classical ML models trained solely on network data, demonstrating significant improvements across all evaluation metrics.

- **Margin Transfer**: Neural Network models incorporating process data-derived learning parameters through margin transfer exhibited superior performance compared to baseline models, highlighting the effectiveness of knowledge transfer from process data.

- **Distillation**: Distilling knowledge from a process data-trained teacher model to network data-trained student models (both DNN and CNN architectures) significantly enhanced detection accuracy, precision, and F1-score.

- **Transfer Learning**: While both pre-trained (using process data) and baseline CNN models showed comparable performance, the pre-trained model exhibited slightly better recall and lower standard deviation, suggesting enhanced robustness.

This section introduces a novel approach to developing robust and accurate Machine Learning (ML)-based Network Intrusion Detection Systems (NIDS) for Industrial Control Systems (ICS) by integrating industrial network traffic and physical process data during the training phase of a supervised learning framework. While the trained model utilizes only network traffic during operation, the incorporation of process data as privileged information during training significantly enhances its detection capabilities, leading to improved accuracy and robustness compared to traditional methods relying solely on network data.

5.6 Multi attack scenario results

We train a single classifier over *multiple* attacks spanning SSSP/SSMP/MSSP/MSMP topologies to test cross-scenario generalisation. Unless noted otherwise, we use the balanced modelling dataset the subset sizes presented in Table 5.2.

5.6.1 Knowledge Transfer Experiment

According to the findings presented in subsection5.3.1 and in (Pordelkhaki et al. 2021), the initial experiment on knowledge transfer shows that the SVM+ (LUPI) algorithm, trained on both network and process features, outperforms baseline ML models trained exclusively on network features. The classification process in this experiment was restricted to the detection of a single type of attack from the SWaT dataset, specifically Attack 36. This experiment expands upon the previously suggested work by examining numerous attacks from the SWaT dataset (outlined in section 3.3). This experiment follows the protocol in Section 5.1.2 an Table 5.2 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.
The training dataset, which includes network and process features, was utilised to train the SVM+ (LUPI) algorithm. Subsequently, the test dataset, comprising solely of network features, was employed to evaluate its performance. The efficacy of the SVM+ (LUPI) algorithm has been analysed by comparing its results with those of the five baseline ML models. These models were trained exclusively on network data and did not incorporate the privileged process data during training. This assessment specifically focused on analysing the performance of K-Nearest Neighbour (K-NN), Logistic Regression (LR), Decision Tree (DT), Multilayer Perceptron (MLP), and Support Vector Machine (SVM) (Géron 2019). Previous researchers have extensively utilised these algorithms to tackle

comparable intrusion detection issues (Goh, Adepu, Tan et al. 2017; Junejo and Goh 2016; Bernieri et al. 2019). The parameters of these models have been optimised through a comprehensive and randomised parameter searches and were tuned through a 5-fold cross-validation on the training set. In order to establish a fair basis for comparing algorithm performance during runtime, each algorithm was executed five times and the average results from these runs were recorded. Unlike the SVM+ (LUPI) algorithm, which considers process features, these models have solely relied on network data for training and testing.

Classifier	Accuracy	Precision	Recall	F1-score
KNN	75.586% (±0.582%)	75.254% (±0.742%)	75.244% (±0.372%)	74.528% (±0.398%)
LR	55.980% (±0.619%)	56.219% (±0.515%)	56.204% (±1.079%)	55.663% (±0.671%)
DT	86.477% (±0.500%)	56.457% (±0.620%)	55.216% (±0.497%)	56.246% (±0.327%)
MLP	63.830% (±0.758%)	65.482% (±3.230%)	64.384% (±3.965%)	64.087% (±1.618%)
SVM	62.767% (±0.430%)	63.708% (±0.387%)	63.079% (±0.305%)	62.219% (±0.837%)
$SVM+_{LUPI}$	69.217% (±0.018%)	93.530% (±0.0101%)	62.807% (±0.0232%)	75.132% (±0.0178%)

Table 5.9: Multi-attack results for SVM+ (LUPI). Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2; metrics section 3.5; label alignment per section 3.4.0.6.

Findings and Analysis

The findings, presented in Table 5.9, indicate that the SVM+ (LUPI) algorithm consistently outperforms other conventional ML algorithms in terms of precision and F1-score, with improvements of 49.19% and 21.47% respectively on average. It should be emphasised that the precision attained by SVM+ in this

experiment far exceeds that of other conventional ML algorithms, underscoring the superiority of the SVM+ algorithm in accurately predicting attack records. Furthermore, the standard deviation of the results achieved from multiple iterations of the SVM+ algorithm, when compared to other Machine Learning algorithms utilised in this experiment, exhibited negligible changes and remained within the range of ±0.01. This serves as evidence of the robust performance of this model across 5 runs.

5.6.2 Margin Transfer Experiment

As outlined in Section 3.6.3, the Margin Transfer approach introduces a novel technique for integrating privileged information into the Neural Network (NN) models. This is achieved by utilising the learning rate of the privileged information-trained model to train a model with the original data. To accomplish this, the steps outlined in previous studies by (Momeni et al. 2018) and (Sharmanska et al. 2013) were used. The findings of the feature importance analysis (as discussed in section 3.6.3) indicate that process data exhibits stronger predictive power when compared to network data. Utilising this finding, the process data was used for training a simple linear classifier with Softmax activation function (Bridle 1989) in the output layer, allowing for the computation of class probabilities. The probabilities obtained from the Softmax function acted as an identifier for the difficulty of the samples; The higher the probability, the easier the example is. The weights obtained from this neural network model were subsequently utilised to train a two-layer neural network with Rectified Linear Unit (ReLU) activation function (Fukushima 1975) and Softmax function in the output layer, using network data. This experiment follows the protocol in Section 5.1.2 an Table 5.2 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.

The neural network model referred to as NN_{MT} has been trained utilising the network features of the training dataset. Additionally, this model integrates the weights derived from a process feature-trained model as an indicator of sample difficult. The $NNbaseline$ model was trained using only the network data incorporating no difficulty measures. The detection capabilities of both neural network models we assessed using the evaluation measures, which are displayed in the Table 5.10. In order to establish a fair comparison, the testing phase was performed five times for each neural network model, and the average findings from these five iterations were documented.

Classifier	Accuracy	Precision	Recall	F1-score
$NN_{baseline}$	50.100% (±0.400%)	50.0% (±0.500%)	50.00% (±0.500%)	49.600% (±0.002%)
NN_{MT}	53.100% (±3.400%)	53.100% (±3.800%)	53.500% (±4.200%)	52.300% (±3.300%)

Table 5.10: Multi-attack results for Margin Transfer based NIDS. Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2; metrics section 3.5; label alignment per section 3.4.0.6.

Findings and Analysis

The experimental results indicate that the NN_{MT} method, which integrates the difficulty measures during the learning phase, outperforms the baseline NN model ($NN_{baseline}$) in all evaluation metrics.

5.6.3 Distillation Experiment

The objective of this experiment is to examine the efficacy of the distillation approach (as discussed in section 3.6.4) in the integration of privileged process data during the learning phase of NIDS for detection of multiple attacks.

This experiment follows the protocol in Section 5.1.2 an Table 5.2 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.

Our initial step involved training a teacher model through the use of privileged process samples. Following that, an analysis was conducted on the performance of two Deep Neural Network (DNN) models, known as student models, which shared a similar structure. The DNN models comprised an input layer with 1000 neurons, in addition to the two hidden layers with 50 and 10 neurons, respectively, with the ReLU activation function and a softmax layer for each class. The Adam optimizer (Kingma et al. 2014) was employed and its learning rate was set to 0.001 using a grid search. Furthermore, the batch size was determined as 1000 and the number of epochs as 10 utilising the grid search technique . Given that the Softmax was employed for calculating the loss function, as stated in section 3.6.4, we passed the non-normalised probabilities from the neural network's last layer to the loss function. This was accomplished by configuring the loss function with the *from-logits* option set to *True*. Both student models were trained with network data, however, the knowledge gained by the teacher model from process data was distilled into the $DNN - Student_D$ while the other DNN student model, denoted here as $DNN - Student_S$, is trained using the network data only. A similar experiment was conducted using Convolution Neural Networks (CNN) model (LeCun et al. 1998), which proved superior performance in similar cyber security detection tasks. Similar to the previous experiment, teacher and student models were structured similarly comprising two 1 Dimensional Convolution layers with the ReLu function, followed by a 1 Dimensional MaxPooling layer, a flattened layer. The training of a CNN-based teacher model was conducted using process data and a CNN model named $CNN - Student_S$ was trained using only network data. In addition, during the

training process, the second student model, denoted as $CNN - Student_D$, acquired the learning pattern from the teacher model by utilizing network data. For a fair comparison, we run the training and testing of each of the DNN and CNN models five times and we reported the average testing results over 5 runs in Table 5.11.

Classifier	Accuracy	Precision	Recall	F1-score
$DNN - Student_S$	45.03% (±0.027%)	44.84% (±0.008%)	70.86% (±0.004%)	54.92% (±0.007%)
$DNN - Student_D$	71.65% (±0.023%)	94.76% (±0.003%)	45.82% (±0.047%)	61.64% (±0.042%)
$CNN - Student_S$	45.66% (±0.034%)	46.62% (±0.036%)	70.84% (±0.017%)	56.18% (±0.029%)
$CNN - Student_D$	70.52% (±0.008%)	93.14% (±0.019%)	44.32% (±0.009%)	60.06% (±0.011%)

Table 5.11: Multi-attack results for Distillation-Based NIDS. Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2; metrics section 3.5; label alignment per section 3.4.0.6.

Findings and Analysis

The experimental results shown in in Table 5.11 demonstrate that the F1-score performance of the DNN-StudentD and CNN-StudentD models, trained using the distillation approach with the privileged process data, surpasses that of the baseline DNN-StudentS and CNN-StudentS models, which are trained solely using network data. Nevertheless, it was observed that the DNN-StudentD and CNN-StudentD models employing the distillation method exhibited lower recall outcomes in comparison to the DNN and CNN models respectively. This result might suggest that these models were not successful in reducing the number of False Negatives.

5.6.4 Transfer Learning Experiment

This experiment investigated the utilisation of the Transfer Learning approach (as elaborated in section 3.6.5) for integrating privileged process data during the learning phase of NIDS to detect multiple attacks. This experiment follows the protocol in Section 5.1.2 an Table 5.2 with label alignment per section 3.4.0.6. Dataset scale and per-experiment subset sizes are summarised at the start of this chapter.

The experiment was planned with consideration of the feature extraction application of transfer learning. The implementation involved the utilisation of a Convolution Neural Network (CNN) model comprising two-Dimensional Convolution layers and a max pooling layer. Subsequently, a Deep Neural Network (DNN) was employed, consisting of an input layer with 1000 neurons, as well as two hidden layers consisting of 100 and 10 neurons respectively, all activated by the ReLU activation function. The model was optimized using a grid search optimisation algorithm. The convolution layers were configured with a filter and kernel size of 5. The Adam optimizer was employed, with a learning rate of 0.001. The batch size and epoch count were set to 1000 and 100, respectively. The CNN model was initially trained on the privilege process data features extracted from the training dataset, which comprised both process and network features. Following the training process, the Convolution layers were frozen and the DNN part of the model was re-trained with network data only. Pre-trained CNN model, denoted here as $pre-trained-CNN$, was tested on the network data only. The performance of this pre-trained model was compared against a similarly structured CNN model , denoted here as CNN, that was trained and tested using network data only. In order to ensure a fair comparison, we conducted five rounds of testing for each of the pre-trained DNN and DNN models. The average testing results over these five runs are presented

in Table 5.12.

Classifier	Accuracy	Precision	Recall	F1-score
CNN	64.500% (±7.900%)	65.200% (±20.500%)	64.500% (±7.600%)	59.500% (±13.600%)
$CNN_{Pretrained}$	64.400% (±0.600%)	65.700% (±1.200%)	64.400% (±0.700%)	63.700% (±1.600%)

Table 5.12: Multi-attack results for Transfer Learning based NIDS. Mean±sd over 5 repeats; train features: network+process (privileged); test features: network-only; evaluation protocol section 5.1.2; metrics section 3.5; label alignment per section 3.4.0.6.

Findings and Analysis

The experimental results show that the Pre-trained CNN and baseline CNN models exhibit almost similar behaviour regarding the accuracy, recall and precision. As presented in Table5.12 although it seems the two models behave almost similarly, the transferred model is performing better with reference to F1-score obtained Reported average results also suggest that the transferred model performance is more stable in each run due to the lower standard deviation calculated for the average results.

5.7 Key findings from multi-attack scenario

This research aimed to consider the key characteristics of ICS and its highly connected network architecture in proposing a new framework for ML-based network intrusion detection in ICS. This requires using both network traffic and process data to train the ML-based NID model however, the model would solely analyse network traffic during runtime. The framework is based on the Machine Learning concept known as LUPI. The validation of this framework

was conducted through the use of tailored techniques involving LUPI concept, which included Knowledge transfer, Margin transfer, Distillation, and Transfer Learning. The key findings and observations derived from the conducted experiments are:

- Despite the superior capabilities of deep learning algorithms, such as CNN, compared to ML algorithms, such as SVM, our research reveals that LUPI using ML methods (as presented in Table. 5.9) achieves better results than the investigated LUPI utilising DNN-based methods. This could potentially be attributed to the limited size of the training data utilised in our ICS application.

- The SVM+ employs a future-oriented teaching approach, which is the rationale behind its superior performance compared to the DNN-based models employed in this study. The reason for this is the influence that each input variable has on the decision made by the model.

- Despite the high performance of LUPI utilising ML methods like SVM+, it falls short in terms of scalability with large datasets. Conversely, CNN and DNN-based models excel in their ability to scale with such datasets. As a matter of fact, the greater the amount of training data provided to the DNN-based models, the greater the accuracy of their predictions.

- Although the LUPI - based models did not consistently outperform their counterpart models in some experiments, the overall performance of the LUPI - based model demonstrated robustness and stability across all experiments. This was evident from the generation of lower standard deviation across all performance metrics compared to their counterpart models.

5.8 Multi-attack scenario : Summary

This section presents a comprehensive evaluation of proposed PINID framework for developing ML-based NIDS specifically designed for ICS. This framework distinguishes itself by incorporating both network traffic data and process data during the training phase, while relying solely on network traffic data during operation, aligning with the practical constraints of real-world NIDS deployments.

The framework leverages the concept of Learning Using Privileged Information (LUPI), which allows ML models to learn from additional information during training that is not available during testing. The SWaT testbed, a scaled-down water treatment plant, and its associated datasets, comprising network traffic and process data from various attack scenarios, were utilized to evaluate the framework's efficacy.

Four distinct ML techniques adapted to LUPI paradigm and were implemented and rigorously evaluated: Knowledge Transfer, Margin Transfer, Distillation, and Transfer Learning. Each technique was assessed based on its ability to enhance the performance of the NIDS model in detecting multiple attack scenarios. The experimental results yielded several key findings. Firstly, despite the advancements in deep learning, LUPI implementations using traditional ML algorithms, particularly the SVM+ algorithm, outperformed those using Deep Neural Networks (DNNs) in this study. This observation is attributed to the limited size of the training data, suggesting that DNNs may require larger datasets to generalize effectively in this context.

Secondly, the SVM+ algorithm demonstrated superior performance compared to other LUPI techniques, attributed to its "future-oriented teaching approach." This approach emphasizes the influence of each input variable on the model's decision-making process, contributing to its enhanced accuracy.

Thirdly, while SVM+ excels in accuracy, it faces scalability challenges with large datasets. Conversely, DNN-based models exhibit superior scalability, achieving higher accuracy as the size of the training data increases.
Finally, despite not consistently outperforming their counterparts in every experiment, LUPI-based models consistently demonstrated robustness and stability across all experiments. This is evidenced by the lower standard deviations observed across all performance metrics compared to their non-LUPI counterparts.
This chapter provides compelling evidence for the potential of the proposed LUPI-based framework in enhancing the accuracy and robustness of NIDS in ICS environments. The study highlights the importance of considering both network traffic and process data during training to develop more effective and reliable intrusion detection systems for critical infrastructure. However, further research is encouraged to address the scalability challenges associated with certain LUPI techniques and to explore the impact of larger datasets on the performance of DNN-based LUPI models.

5.9 Implications of this research for ICS security

Our results reinforce a practical pattern for defenders: Train models to be process-aware, but run them where teams already have control—on the network. In OT/ICS, this lines up with guidance to prioritise visibility and segmentation at Purdue Levels 1–3, where most commands and telemetry flow and where passive monitoring carries the least production risk (Stouffer et al. 2023). For environments that can't host heavy analytics on controllers or historians, a NIDS that was trained to understand expected process effects is a low-friction way to catch command misuse earlier(e.g. Impair Process Control,

TA0106) (MITRE Corporation 2025h).

A second implication concerns alert quality and operator trust. Adding short guard-bands around attack boundaries (our label-alignment step) reliably cut false positives from timing variances, lifting precision without hurting recall. In practice, this mirrors how security/control-room teams already use alarm times and correlation windows to filter noise while keeping time-critical events. Building this alignment into model evaluation—and later into alarm logic—helps ensure new analytics don't add noise to safety-critical workflows that already dealing with process, maintenance, and cyber alarms (Stouffer et al. 2023).

Third, the approach complements—not replaces—defence-in-depth. IEC/ISA 62443 stresses basics like asset inventory, segmentation, allow-listing, and secure remote access. A NIDS that understands command semantics (function code, target tag, requested values/limits) can sit alongside these controls to catch threats that slip past coarse policy—supporting earlier detection of abnormal command intent and faster containment (International Society of Automation (ISA) 2025).

Finally, before enabling blocking or automated responses, models should be validated in a staging environment that mirrors the historian/network alignment assumptions in the training stage of PINID framework. Tune thresholds to the environment's real base rates and track precision, recall, and balanced accuracy to avoid optimism from training-time class balance.

Chapter 6

Discussion and finding evaluation

The findings of this research have been organized and presented in Chapter 5. This chapter is specifically devoted to evaluating the hypotheses outlined in Section 1.2, based on the findings presented in earlier chapters. this will include the analysis of the findings regarding the application of ML techniques used in this research. The assessment of the study's design and the presentation of limitations and potential strategies for improving future research designs are discussed in Section 6.2. this section also includes the assessment of the findings in in light of the identified limitations and the feedback received from peers and domain experts.

6.1 Hypotheses and the Performance Assessment

This section reflects on the results presented in Chapter 5 and Chapter 6 of this study in the context of the research Hypotheses, presented in 1.2, assessing whether this our hypothesis was proved true by the results obtained.

Hypothesis: Incorporating process information along the network traffic during training of a supervised ML-based NIDS following proposed PINID framework improve the performance of the ML-based NIDS in detecting control attacks in ICS network when tested on the network traffic only.

6.1.1 Single Attack Scenario

To evaluate our hypothesis we have used the data available from SWaT testbed and designed an experiment as a proof of concept to assess the effectiveness of the proposed framework focusing on detecting a single attack in the SWaT ICS network (attack 30). To do so an SVM model (denoted as SVM+) was trained following LUPI framework with a data set consist network traffic and process information. Accuracy, Precision, Recall and F1 score has been used as performance metrics to compare the performance of this model with classic ML algorithms which were trained and tested using network traffic only. The initial result obtained from this experiment suggested that on average, the accuracy, precision, Recall and F1-scores obtained by SVM+ outperform the results obtained by other classical ML algorithms used in this experiment. Although the accuracy obtained by SVM+ outperforms all baseline ML algorithms, it was slightly lower that the accuracy obtained by DT classifier as presented in Table 5.5. However SVM+ presented bettre performance in compare to DT Classifier with respect to precision, recall and F1 score. Additionally, SVM+ demonstrated the lowest standard deviations in all four performance metrics across five run of training/test in compare to other classifiers, which was interpreted as reliability and robustness metric of this model.

To assess the proposed framework further, a NN technique known as Margin Transfer was used to train a model following the proposed framework (denoted as $NN(1HL)_{PI}$). The performance of this model was then compared against

similar model which was solely trained and tested on network traffic. as presented in Table 5.6, this model outperformed similar model that was trained and tested following a classic classification methodology. BY improving the structure of the $NN(1HL)_{PI}$ model from a NN model with one hidden layer to a NN model with two hidden layer, denoted as $NN(2HL)_{PI}$, the obtained results outperform the initial model as well as the results obtained by SVM+ model with lower standard deviation across all the performance metrics over five run of train and test. focusing on NN techniques, we have adapted the Distillation technique to the proposed PINID framework and created a DNN and a CNN student model , known as distilled student models. While training with Network traffic, these distilled student models received the distilled information from similarly constructed teacher models (DNN and CNN) that were trained on process data. These models were then tested using network traffic and their performance was compared with similarly structured student models which were trained and tested on network traffic. As presented in Table 5.7 the distilled student models outperform the similarly structured student models with distilled CNN student model outperform the distilled DNN student model on all performance metrics except for the precision.

Transfer learning technique was also explored following PINID framework. We created trained a CNN model including DNN layers. The CNN model was first trained using process data. Following the initial training, the CNN layers of the model were frozen and the DNN layers of the model were re-trained using network traffic. The final model was then tested using network traffic. The performance of this model was compared with a baseline model that was similarly structured CNN model which was trained and tested on network data. Both model performed equally poor in identifying selected attack. We believe for the pre-trained model to be able to perform at its possible best it should be

initially trained with considerably large dataset. In this experiment the the size of the process data used for initial training was limited to the existing dataset.

Based on the findings from the performance of SVM+ , Margin Transfer and Distillation technique following the proposed framework in identifying a single , we can accept the hypothesis. However, the performance of the proposed framework needs further evaluation to access the generalisation of the proposed framework. Therefore, similar experiments were performed on a dataset including a subset of attacks as presented in Table 3.4 and discussed in 3.3 the selected dataset included 5 attacks which as a representative of the 4 attack types considered in development of the SWaT dataset.

6.1.2 Multiple Attack Scenario

The performance of SVM+ in detecting a various attack types in compare to the other classic classifiers was shown in Table 5.9. Obtained results demonstrate that the SVM+ outperform all the conventional classifiers in terms of Precision and F1-score. The precision score attained by SVM+ algorithm exceeded that of other classic ML algorithms by far. This demonstrate the superiority of SVM+ in predicting attack records. Additionally, similar to the previous experience with the SVM+ algorithm, the negligible standard deviation in the results obtained from SVM+ performance metrics over five train and test run in compare to the other algorithm is an evidence to the reliability and robustness of this model.

In a similar experience the margin transfer technique was examined in detecting of a set of different attacks following proposed framework. The results of this experiment which is shown in Table 5.10 present slight improvement in performance metrics in compare to a similar model which was trained and tested on network traffic.

Same experiment was setup to evaluate the performance of distillation tech-

nique in detection of multiple attack type. The findings from this experiment which was presented in Table 5.11 demonstrate that the DNN distilled student and the CNN distilled student outperform the standard student in accurately identifying the attacks over all metrics except for the recall. It is important to highlight That the performance of both distilled student in detecting the attacks with respect to Precision score exceeded the standard counterpart models. The DNN distilled student average performance outperform SVM+ model in identifying multiple attack types.

Transfer learning technique was reassessed with respect to the identifying multiple attacks following PINID framework. Pre-trained and baseline models were created using the multi-attack database. findings from this experiment is demonstrated in Table 5.12. The performance of these model is slightly improved when compared to the similar experiment with single attack database, which cam be due to the larger dataset used for training. Although the results obtained from pre-trained and baseline models demonstrated similar performance in detecting different type of attacks, the pre-trained model demonstrate considerably lower standard deviation across all performance metrics over five run of train and test. this is an evidence to the robustness of the model yet the performance with respect to accuracy, precision, recall and F1-score needs further improvement.

The performance of the ML-based NIDS that were trained following the proposed PINID framework when compared to the ML-based NIDS that were train and tested on network traffic, present improvement specially with respect to the accurate identification of attack records is evident in the high precision score attained by these model except for the transfer learning approach. Furthermore, all the models trained following PINID framework demonstrated lower standard deviation across all performance metrics over five train and test run which

is another evidence of the improved performance of these models. A practical path for asset owners is to train with process data (historian snapshots as privileged data) but deploy a NID. This keeps change-control light while improving sensitivity to command misuse (e.g. unauthorised writes) which can look normal at packet level unless judged against expected process effects. In short, the NID runs on the network where the SOC already has visibility, but it has been trained to understand what a legitimate command should do to the plant.

6.2 Study Design: Limitations and Lessons Learned

There were a number of limitations in this research design, which will be explored in this section, based on which we suggest approaches for designing future studies in this space to overcome the shortcomings of this study.

6.2.1 Research Structure and Process

It is worth noting that there are potential limitations to consider in this study. The model's training features are carefully chosen by considering common instances identified in studies, and guidance provided by the test bed and data set provider. In this research, all the features from both the SWaT process and network datasets were utilised. As a possible outcome, these experiments could have led to the development of models that were both heavier and slower in nature.

Throughout the development and evaluation process of the framework, we placed special emphasis on analysing the Brute Force IO and Unauthorised Command Message adversarial techniques. The potential impact of this could extend to the framework's ability to effectively identify and address various adversarial techniques that specifically target ICS.

6.2.2 Study Setup

We ensured that the research setup was as realistic as possible with respect to the test-bed and dataset used. This has limited this study to the SWaT dataset which was collected from a small scale yet operational water treatment plant with a distributed control systems use and data flow alignment to the Perdue model. This limited the evaluation of the proposed framework to the SWaT dataset.

6.2.3 Data quality.

Using the boundary guard-bands (section 5.2) improved Precision with almost no change in Recall. This tells us that small timing shifts at the start and end of attacks were boosting false positives, a data quality issue. To address this, we didn't change any labels; we trimmed short slices at the boundaries to reduce the mismatch between labels and behaviour (section 3.4.0.6). That said, very slow, stealthy attacks could still sit partly inside the guard-band. This should be confirm this with cleaner, more precise ground-truth data.

6.2.4 Generalisability.

In single-attack tests, the models fit each scenario very well. In the multi-attack setting, the model's accuracy decreased a little as per-scenario but performed better with respect to generalising across scenarios within SWaT's SSSP/ SSMP/ MSSP/ MSMP topologies (section 3.2.3). This shows the approach works on SWaT, but this still need to be check how well it transfers to other plants with different processes, safety interlocks, and class balances.

6.2.5 Legacy dataset.

All our experiments used SWaT, a water-treatment ICS with its own control loops and safety interlocks. In the multi-attack runs, training one model across SSSP/ SSMP/ MSSP/ MSMP improved generalisation across scenarios within SWaT, but we have not tested this in other settings (e.g. power, chemicals), so we cannot claim it generalises there yet. The approach works best when command details (function code, tag, requested limits) map cleanly to well-constrained plant physics. That holds in SWaT and likely in many regulated processes, but it still needs checking elsewhere. To mitigate this, we focused on stage-agnostic command misuse (e.g. write/force actions); nevertheless, independent validation on additional testbeds remains necessary.

Chapter 7

Conclusions and future work

7.1 The Big Picture

We have presented the Process Informed Network Intrusion Detection framework for ICS, an ML-Based NIDS which breaks new ground i.e. unlike other proposed approaches in this research domain that are focused on data captured from networks or processes in an ICS environment, the proposed framework focuses on the main characteristics of ICS in controlling a physical process through a high network architecture. We achieve this by utilising valuable process-level information during the training phase of the ML-Based NIDS, however, utilising network traffic alone during the run-time. We evaluated this framework leveraging LUPI paradigm through Transfer Learning, SVM+ and Distillation techniques and utilising SWaT dataset for detecting Brute Force and Unauthorised Command Message attacks aiming to Impair Process Control. Our first experiment on Knowledge Transfer demonstrates SVM+ trained on integrated data via the LUPI framework, outperforms baseline models trained on network features on F1-score and precision metrics. Our experiment on Margin transfer suggests when the learning difficulty – obtained from a model trained

with privileged information – is transferred to a neural network model during the training process with the network data, all performance metrics present slight improvement in comparison with the similar neural network trained with network data only. Our experiment on distilling knowledge from a teacher model trained with process data – as privileged information – to a student model while training with network data outperforms a student model trained with network data only.

7.1.1 Re-stating research objectives

Objective 1 : Develop an ML-based NIDS for ICS that incorporates process and network data. (*Achieved.*)
We designed PINID framework in which process data is used as privileged information for training an ML-based NIDs while the deployed detector is running on network daya only. Across the single- and multi-attack scenarios, the framework delivered strong Precision/Recall and F1 under the common protocol (see the results tables in sections 5.3 and 5.6).

Objective 2: Evaluate the practicality of LUPI in ICS. (*Achieved.*)
SVM+ trained with network + process features (privileged) and tested network-only outperformed network-only baselines in the single-attack series, and the advantage persisted in the multi-attack scenarios (sections 5.3 and 5.6). This shows LUPI is practical: it improves accuracy while keeping runtime simple (network-only).

Objective 3: Compare traditional NIDS vs LUPI-based NIDS on single and multi attacks. (*Achieved.*)
LUPI-based models performed better than the standard baselines throughout. In single-attack tests, they showed the biggest improvements for each scenario. In multi-attack tests, a single model lost a little per-scenario accuracy but gen-

eralised better across different scenarios within SWaT (sections 5.3 and 5.6).

Objective 4: Evaluate robust neural-network knowledge transfer following LUPI. (*Achieved.*)
Distillation, margin transfer and Transfer learning were evaluated and Distillation (teacher trained with process context; student trained network-only) presented a considerable improvement over their network-only counterparts and were most stable when trained on the larger, consistent subsets (5.3 and 5.6).

7.2 Summary of Contribution

Even though there is already a substantial body of research on ML-based anomaly detection for the SWaT test-bed, it is important to highlight that this study is the first to utilise LUPI for ML-based network intrusion detection systems in the ICS domain. The major contributions of this research can be outlined:

- This research introduces PINID, a novel Network Intrusion Detection System (NIDS) framework specifically designed for ICS environments. PINID distinguishes itself by considering both cyber and physical aspects of the system, offering a more comprehensive approach to security. While existing research primarily focuses on network data for intrusion detection, PINID leverages the combined strengths of both network and process data to enhance the detection of cyberattacks within an ICS. This integration enables a more holistic and accurate assessment of the ICS security posture. In line with established practices, this research utilises network data for developing an ICS-specific NIDS, adhering to the principles of the LUPI framework. However, unlike many proposed techniques that disregard the valuable information embedded within process data, this ap-

proach retains and leverages this knowledge. This results in a more robust and reliable Machine Learning (ML)-based NIDS specifically tailored for ICS environments.

- The PINID framework's effectiveness was rigorously evaluated against a range of attacks targeting industrial processes, including brute force IO and unauthorised command messages, aligned with MITRE ATT&CK for ICS and considered hierarchical network architecture of the Purdue Model. The evaluation used the SWaT dataset, which comprises ICS experiments on a small-scale water treatment plant and represents 36 different attack types, to assess attacks on single and multiple field devices within a simulated water purification process. The study also highlighted the importance of data labelling for supervised learning in ICS security and proposed an approach to address limitations in existing datasets.

- In a single attack scenario, the PINID framework's performance was evaluated using common supervised machine learning algorithms: SVM, LR, DT, MLP, KNN. Network traffic data was used for both training and testing phases.The study compared the performance of these algorithms against a knowledge transfer technique, SVM+, an adaptation of the SVM algorithm incorporating Learning Using Privileged Information (LUPI). In this approach, process information supplemented network traffic data, enhancing the algorithm's learning capabilities. Furthermore, the framework's evaluation included robust neural network methods for single attack detection. This validation encompassed assessing margin transfer and distillation techniques alongside transfer learning.

- Given the complexity of multi-attack scenarios and their diverse characteristics, the PINID framework's performance was rigorously assessed

in this context. Common supervised machine learning (ML) algorithms, including SVM, LR, DT, MLP, and KNN, were employed, utilising network traffic data for both training and testing phases. The study specifically compared these algorithms' performance against SVM+, a knowledge transfer technique. which adapts the SVM algorithm by incorporating Learning Using Privileged Information (LUPI), appending process information to network traffic data as supplementary information. Beyond traditional ML methods, the framework evaluation encompassed robust neural network approaches. This included assessing margin transfer and distillation techniques, in addition to transfer learning, all implemented within a neural network structure.

Originality: We believe this is the first time Learning Using Privileged Information has been applied to a NIDS in ICS. In short: we train with process context, but we run the model using network data only in runtime. This mix—process-aware learning with a light, low-friction deployment—drives the improvements reported in sections 5.3 and 5.6.

Practical implications: Two points carry straight into operations. First, adding short guard-bands at attack boundaries (our label-alignment step) cut false positives without hurting recall (section 5.1.2). Second, training with process context improved precision on command misuse (e.g. unauthorised writes) while keeping the live system network-only (sections 5.3 and 5.6). Together, these changes mean earlier and more trustworthy alerts for security teams.

7.3 Future Work

This research highlights a framework that can potentially enhance NIDS performance in ICS and other Cyber-Physical Systems like the Internet of Things

(IoT). Future improvements could involve:

- Various ICS datasets : The evaluation of this framework can leverage ICS datasets derived from diverse industrial environments, employing a hierarchical network architecture. This approach enables the validation of the framework's efficacy across a spectrum of process types, including continuous, discrete, and batch processes, each characterized by distinct process data characteristics. Furthermore, this methodology facilitates a comprehensive assessment of the proposed framework's resilience against a variety of cyberattacks targeting different industrial and proprietary communication protocols.

- Feature selection: The integration of optimised feature selection methodologies into this framework, specifically within the context of ICS network intrusion detection, presents a significant area for further investigation. A comprehensive assessment of such integration could encompass its impact on multiple facets of the NIDS, including but not limited to; performance enhancements, improvements in generalisation capabilities, and the implications for computational cost and overall scalability.

- Control Data : The integration of auxiliary data from ICS, including but not limited to alarm and event logs, and control set-point values, constitutes a promising area of research in the context of NIDS enhancement. The rationale for this proposition emerge from the capacity of such data to provide a comprehensive insight of the normal operational state within the ICS environment.

- Integrating Adversarial knowledge: While the present framework demonstrates efficacy in detecting a variety of attack vectors, further research is needed to explore the integration of adversarial knowledge. Specifically,

incorporating information about adversarial Tactics, Techniques, and Procedures (TTPs), such as Indicators of Compromise (IOC)s and privilege escalation methods, presents a valuable opportunity to rigorously assess and potentially improve the framework's overall performance.

- Diverse neural network architecture and optimisation: With the advancements in artificial intelligence and the various configurations of neural network structures, future research in this field could explore the implementation of alternative neural network architectures, building upon the techniques employed in this study.

- Unsupervised learning: Given the challenge of obtaining labelled datasets from industrial environments, evaluating the performance accuracy of the proposed framework can be done using an unsupervised approach in ICS network intrusion detection.

References

Abdelaty, Maged et al. (2021). 'DAICS: A deep learning solution for anomaly detection in industrial control systems'. In: *IEEE Transactions on Emerging Topics in Computing* 10.2, pp. 1117–1129.

Abid, Aymen et al. (2017). 'Outlier detection for wireless sensor networks using density-based clustering approach'. In: *IET Wireless Sensor Systems* 7.4, pp. 83–90.

Addeen, Hajar Hameed et al. (2021). 'A survey of cyber-physical attacks and detection methods in smart water distribution systems'. In: *IEEE Access* 9, pp. 99905–99921.

Agha, Adeel (2024). 'Bolstering the Cyber Defenses of Critical Infrastructure: An In-Depth Analysis of AI-Driven Security for Industrial Control Systems'. In.

Aghdam, Mehdi Hosseinzadeh et al. (2016). 'Feature selection for intrusion detection system using ant colony optimization.' In: *Int. J. Netw. Secur.* 18.3, pp. 420–432.

Agrawal, Anand et al. (2018). 'Poster: Physics-based attack detection for an insider threat model in a cyber-physical system'. In: *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 821–823.

Ahmad, Iftikhar et al. (2014). 'Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components'. In: *Neural computing and applications* 24, pp. 1671–1682.

Alexander, Otis et al. (2020). 'MITRE ATT&CK for Industrial Control Systems: Design and Philosophy'. In: *The MITRE Corporation: Bedford, MA, USA*.

Almalawi, Abdulmohsen, Adil Fahad et al. (2016). 'An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems'. In: *IEEE Transactions on Information Forensics and Security* 11.5, pp. 893–906. DOI: 10.1109/TIFS.2015.2512522.

Almalawi, Abdulmohsen, Xinghuo Yu et al. (2014). 'An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems'. In: *Computers & Security* 46, pp. 94–110.

Alves, Thiago et al. (2018). 'Embedding encryption and machine learning intrusion prevention systems on programmable logic controllers'. In: *IEEE Embedded Systems Letters* 10.3, pp. 99–102.

Amin, Saurabh et al. (2012). 'Cyber security of water SCADA systems—Part I: Analysis and experimentation of stealthy deception attacks'. In: *IEEE Transactions on Control Systems Technology* 21.5, pp. 1963–1970.

Amrouch, Hussam et al. (2017). 'Emerging (un-) reliability based security threats and mitigations for embedded systems: Special session'. In: *Proceedings of the 2017 International Conference on Compilers, Architectures and Synthesis for Embedded Systems Companion*, pp. 1–10.

Anbalagan, Arivazhagan et al. (2025). 'Lightweight CNC digital process twin framework: IIoT integration with open62541 OPC UA protocol'. In: *Production & Manufacturing Research* 13.1, p. 2544981.

Ani, Uchenna P Daniel et al. (2017). 'Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective'. In: *Journal of Cyber Security Technology* 1.1, pp. 32–74.

Aouedi, Ons et al. (2022). 'Federated semisupervised learning for attack detection in industrial internet of things'. In: *IEEE Transactions on Industrial Informatics* 19.1, pp. 286–295.

Ariharan, V et al. (2019). 'Machine learning quorum decider (MLQD) for large scale IoT deployments'. In: *Procedia Computer Science* 151, pp. 959–964.

Ashfaq, Rana Aamir Raza et al. (2017). 'Fuzziness based semi-supervised learning approach for intrusion detection system'. In: *Information sciences* 378, pp. 484–497.

Aslam, Muhammad Muzamil et al. (2024). 'An improved autoencoder-based approach for anomaly detection in industrial control systems'. In: *Systems Science & Control Engineering* 12.1, p. 2334303.

Åström, Karl Johan et al. (2021). *Feedback systems: an introduction for scientists and engineers*. Princeton university press.

Aykut, Ercan et al. (2025). 'Remote Control and Monitoring Protocols for PLC and HMI Systems: A Case Study of MODBUS, WMI and VNC'. In: *2025 9th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*. IEEE, pp. 1–6.

Bansal, Payal et al. (2024). *Artificial intelligence and communication techniques in Industry 5.0*. CRC Press.

Baralis, Elena et al. (2007). 'A lazy approach to associative classification'. In: *IEEE Transactions on Knowledge and Data Engineering* 20.2, pp. 156–171.

Barto, Andrew G et al. (1997). *Introduction to reinforcement learning*.

Bernieri, Giuseppe et al. (2019). 'Evaluation of machine learning algorithms for anomaly detection in industrial networks'. In: *2019 IEEE International Symposium on Measurements & Networking (M&N)*. IEEE, pp. 1–6.

Bhattacharjee, Shameek et al. (2018). 'Towards fast and semi-supervised identification of smart meters launching data falsification attacks'. In: *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 173–185.

Blount, Joseph (June 2021). *Threats to Critical Infrastructure: Examining the Colonial Pipeline Cyber Attack*. URL: https://www.hsgac.senate.gov/hearings/threats-to-critical-infrastructure-examining-the-colonial-pipeline-cyber-attack/.

Boateng, Emmanuel Aboah et al. (2022). 'Anomaly detection for a water treatment system based on one-class neural network'. In: *IEEE access* 10, pp. 115179–115191.

Breiman, Leo (2001). 'Random forests'. In: *Machine learning* 45, pp. 5–32.

Bridle, John (1989). 'Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters'. In: *Advances in neural information processing systems* 2.

Bukharev, DA et al. (2022). 'Application of the technology of hierarchical cluster analysis for clustering the data of information processes of ICS exposed to cyberattacks'. In: *SMART Automatics and Energy: Proceedings of SMART-ICAE 2021*. Springer, pp. 487–496.

Carcano, Andrea, Alessio Coletta et al. (2011). 'A multidimensional critical state analysis for detecting intrusions in SCADA systems'. In: *IEEE Transactions on Industrial Informatics* 7.2, pp. 179–186.

Carcano, Andrea, Igor Nai Fovino et al. (2010). 'State-based network intrusion detection systems for SCADA protocols: a proof of concept'. In: *Critical In-*

formation Infrastructures Security: 4th International Workshop, CRITIS 2009, Bonn, Germany, September 30-October 2, 2009. Revised Papers 4. Springer, pp. 138–150.

Cárdenas, Alvaro A et al. (2011). 'Attacks against process control systems: risk assessment, detection, and response'. In: *Proceedings of the 6th ACM symposium on information, computer and communications security*, pp. 355–366.

Case, Defense Use (2016). 'Analysis of the cyber attack on the Ukrainian power grid'. In: *Electricity Information Sharing and Analysis Center (E-ISAC)* 388, pp. 1–29.

Caselli, Marco, Emmanuele Zambon and Frank Kargl (2015). 'Sequence-aware intrusion detection in industrial control systems'. In: *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, pp. 13–24.

Caselli, Marco, Emmanuele Zambon, Jonathan Petit et al. (2015). 'Modeling message sequences for intrusion detection in industrial control systems'. In: *Critical Infrastructure Protection IX: 9th IFIP 11.10 International Conference, ICCIP 2015, Arlington, VA, USA, March 16-18, 2015, Revised Selected Papers 9*. Springer, pp. 49–71.

Celik, Z Berkay et al. (2018). 'Detection under Privileged Information'. In: *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 199–206.

Çelik, Mete et al. (2011). 'Anomaly detection in temperature data using DBSCAN algorithm'. In: *2011 international symposium on innovations in intelligent systems and applications*. IEEE, pp. 91–95.

Chawla, Nitesh V et al. (2002). 'SMOTE: synthetic minority over-sampling technique'. In: *Journal of artificial intelligence research* 16, pp. 321–357.

Chen, Yongle et al. (2022). 'Cross-domain industrial intrusion detection deep model trained with imbalanced data'. In: *IEEE Internet of Things Journal* 10.1, pp. 584–596.

Cheung, Steven et al. (2007). 'Using model-based intrusion detection for SCADA networks'. In: *Proceedings of the SCADA security scientific symposium*. Vol. 46. Citeseer, pp. 1–12.

Chong, Michelle S et al. (2019). 'A tutorial introduction to security and privacy for cyber-physical systems'. In: *2019 18th European Control Conference (ECC)*. IEEE, pp. 968–978.

Colbert, Edward et al. (2016). 'A process-oriented intrusion detection method for industrial control systems'. In: *International Conference on Cyber Warfare and Security*. Academic Conferences International Limited, p. 497.

Conti, Mauro et al. (2021). 'A Survey on Industrial Control System Testbeds and Datasets for Security Research'. In: *arXiv*. eprint: 2102.05631. URL: https://arxiv.org/abs/2102.05631.

Cordero, Carlos Garcia et al. (2019). 'On generating network traffic datasets with synthetic attacks for intrusion detection'. In: *arXiv*. eprint: 1905.00304. URL: https://arxiv.org/abs/1905.00304.

Cortes, Corinna et al. (1995). 'Support-vector networks'. In: *Machine learning* 20.3, pp. 273–297.

Critical Infrastructure Systems Lab (2024). *BATADAL 2.0 dataset (process + network via DHALSIM)*. URL: https://galelli.cee.cornell.edu/news/batadal-2-0-dataset/ (visited on 07/09/2025).

Curtis, Ken (2005). 'A DNP3 protocol primer'. In: *DNP User Group* 2005.

Dehlaghi-Ghadim, Alireza et al. (2023). 'Anomaly Detection Dataset for Industrial Control Systems (ICS-Flow)'. In: *arXiv*. eprint: 2305.09678. URL: https://arxiv.org/abs/2305.09678.

Demertzis, Konstantinos et al. (2020). 'Gryphon: a semi-supervised anomaly detection system based on one-class evolving spiking neural network'. In: *Neural Computing and Applications* 32, pp. 4303–4314.

Dev, Aditya et al. (2024). 'Enhancing Intrusion Detection Systems through Federated Learning and Gated Recurrent Units'. In: *2024 IEEE Conference on Engineering Informatics (ICEI)*. IEEE, pp. 1–8.

Al-Dhaheri, Mohammed et al. (2022). 'Detection of cyber attacks on a water treatment process'. In: *IFAC-PapersOnLine* 55.6, pp. 667–672.

Dong, Rui-Hong et al. (2018). 'Traffic Characteristic Map-based Intrusion Detection Model for Industrial Internet.' In: *Int. J. Netw. Secur.* 20.2, pp. 359–370.

Fan, Xiaohe et al. (2015). 'Overview of cyber-security of industrial control system'. In: *2015 international conference on cyber security of smart cities, industrial control system and communications (SSIC)*. IEEE, pp. 1–7.

Faramondi, Luca et al. (2021). 'A Hardware-in-the-Loop Water Distribution Testbed Dataset for Cyber-Physical Security Testing'. In: *IEEE Access* 9, pp. 122385–122396. DOI: 10.1109/ACCESS.2021.3109465.

Feng, Ming et al. (2017). 'Deep reinforecement learning based optimal defense for cyber-physical system in presence of unknown cyber-attack'. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–8.

Ferling, Benedikt et al. (2018). 'Intrusion detection for sequence-based attacks with reduced traffic models'. In: *Measurement, Modelling and Evaluation of Computing Systems: 19th International GI/ITG Conference, MMB 2018, Erlangen, Germany, February 26-28, 2018, Proceedings 19*. Springer, pp. 53–67.

Fernández, Santiago et al. (2007). 'An application of recurrent neural networks to discriminative keyword spotting'. In: *International conference on artificial neural networks*. Springer, pp. 220–229.

Fouad, Shereen (2013). 'Metric learning for incorporating privileged information in prototype-based models'. PhD thesis. University of Birmingham.

Fouad, Shereen, Peter Tino et al. (2013). 'Incorporating privileged information through metric learning'. In: *IEEE transactions on neural networks and learning systems* 24.7, pp. 1086–1098.

Fouad, Shereen and Peter Tiňo (2013). 'Ordinal-based metric learning for learning using privileged information'. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8.

Friedman, Nir et al. (1997). 'Bayesian network classifiers'. In: *Machine learning* 29, pp. 131–163.

Fukushima, Kunihiko (1975). 'Cognitron: A self-organizing multilayered neural network'. In: *Biological cybernetics* 20.3, pp. 121–136.

Gao, Wei et al. (2010). 'On SCADA control system command and response injection and intrusion detection'. In: *2010 eCrime Researchers Summit*. IEEE, pp. 1–9.

Gao, Zijun et al. (2024). 'Learning using privileged information with logistic regression on acute respiratory distress syndrome detection'. In: *Artificial Intelligence in Medicine* 156, p. 102947.

Gauthama Raman, M.R. et al. (2020). 'Deep autoencoders as anomaly detectors: Method and case study in a distributed water treatment plant'. In: *Computers Security* 99, p. 102055. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2020.102055. URL: https://www.sciencedirect.com/science/article/pii/S016740482030328X.

Géron, Aurélien (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Ghaeini, Hamid Reza et al. (2016). 'Hamids: Hierarchical monitoring intrusion detection system for industrial control systems'. In: *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, pp. 103–111.

Giallanza, Tyler et al. (2019). 'Keyboard snooping from mobile phone arrays with mixed convolutional and recurrent neural networks'. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.2, pp. 1–22.

Goh, Jonathan, Sridhar Adepu, Khurum Nazir Junejo et al. (2016). 'A dataset to support research in the design of secure water treatment systems'. In: *International Conference on Critical Information Infrastructures Security*. Springer, pp. 88–99.

Goh, Jonathan, Sridhar Adepu, Marcus Tan et al. (2017). 'Anomaly detection in cyber physical systems using recurrent neural networks'. In: *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, pp. 140–145.

Goodfellow, Ian et al. (2017). 'Deep learning (adaptive computation and machine learning series)'. In: *Cambridge Massachusetts*, pp. 321–359.

Guyon, Isabelle et al. (2002). 'Gene selection for cancer classification using support vector machines'. In: *Machine learning* 46.1, pp. 389–422.

Hadeli, Hadeli et al. (2009). 'Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration'. In: *2009 IEEE Conference on Emerging Technologies & Factory Automation*. IEEE, pp. 1–8.

Hadžiosmanović, Dina et al. (2014). 'Through the eye of the PLC: semantic security monitoring for industrial processes'. In: *Proceedings of the 30th Annual Computer Security Applications Conference*, pp. 126–135.

Hasan, Md Al Mehedi et al. (2014). 'Support vector machine and random forest modeling for intrusion detection system (IDS)'. In: *Journal of Intelligent Learning Systems and Applications* 6.01, pp. 45–52.

Hindy, Hanan et al. (2018). 'A taxonomy and survey of intrusion detection system design techniques, network threats and datasets'. In.

Hink, Raymond C Borges et al. (2014). 'Machine learning for power system disturbance and cyber-attack discrimination'. In: *2014 7th International symposium on resilient control systems (ISRCS)*. IEEE, pp. 1–8.

Hinton, Geoffrey et al. (2015). 'Distilling the knowledge in a neural network'. In: *arXiv preprint arXiv:1503.02531* 2.7.

Hobbs, Allegra (2021). 'The colonial pipeline hack: Exposing vulnerabilities in us cybersecurity'. In: *SAGE Business Cases*. SAGE Publications: SAGE Business Cases Originals.

Holdbrook, Raphael et al. (2024). 'Network-Based Intrusion Detection for Industrial and Robotic Systems: A Survey'. In: *Electronics* 13.22, p. 4440. DOI: 10.3390/electronics13224440. URL: https://www.mdpi.com/2079-9292/13/22/4440.

Hong, Junho et al. (2014). 'Detection of cyber intrusions using network-based multicast messages for substation automation'. In: *ISGT 2014*. IEEE, pp. 1–5.

Hou, Chongyuan et al. (2012). 'A probabilistic principal component analysis approach for detecting traffic anomaly in industrial networks'. In: *Journal of Xi'an Jiaotong University* 46.2.

Huda, Shamsul, Jemal Abawajy et al. (2019). 'Automatic extraction and integration of behavioural indicators of malware for protection of cyber–physical networks'. In: *Future generation computer systems* 101, pp. 1247–1258.

Huda, Shamsul, Suruz Miah et al. (2017). 'Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data'. In: *Information Sciences* 379, pp. 211–228.

Hussain, Arif et al. (2024). 'Anomaly Detection using Bi-Directional Long Short-Term Memory Networks for Cyber-Physical Electric Vehicle Charging Stations'. In: *IEEE Transactions on Industrial Cyber-Physical Systems*.

Ieracitano, Cosimo et al. (2020). 'A novel statistical analysis and autoencoder driven intelligent intrusion detection approach'. In: *Neurocomputing* 387, pp. 51–62.

Illy, Poulmanogo et al. (2022). 'A hybrid multistage DNN-based collaborative IDPS for high-risk smart factory networks'. In: *IEEE Transactions on Network and Service Management* 19.4, pp. 4273–4283.

IMPACT Cyber Trust (2015). *ICS Cyber Attack Gas Pipeline Datasets (labeled RTU telemetry streams)*. URL: https://www.impactcybertrust.org/dataset_view?idDataset=1322 (visited on 07/09/2025).

Inoue, Jun et al. (2017). 'Anomaly detection for a water treatment system using unsupervised machine learning'. In: *2017 IEEE international conference on data mining workshops (ICDMW)*. IEEE, pp. 1058–1065.

International Society of Automation (ISA) (2025). *ISA/IEC 62443 Series of Standards*. Overview of the ISA/IEC 62443 series and foundational requirements perspective. URL: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards (visited on 05/10/2025).

Islam, Md Tawfiqul (2025). 'Adversarial Defence Mechanisms In Neural Networks For ICS Fault Tolerance: A Comparative Analysis'. In: *ASRC Procedia: Global Perspectives in Science and Scholarship* 1.01, pp. 404–431.

iTrust, SUTD (2016). *Dataset Info — SWaT (historian CSV and PCAP files)*. URL: https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/ (visited on 07/09/2025).

— (2017). *WADI — Water Distribution Testbed*. URL: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_wadi/ (visited on 07/09/2025).

Al-Jarrah, Omar et al. (2015). 'Network intrusion detection system using neural network classification of attack behavior'. In: *Journal of Advances in Information Technology Vol* 6.1.

Javaid, Ahmad et al. (2016). 'A deep learning approach for network intrusion detection system'. In: *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 21–26.

Jiang, Jehn-Ruey et al. (2022). 'Industrial control system anomaly detection and classification based on network traffic'. In: *IEEE Access* 10, pp. 41874–41888.

Joshi, Chaitali et al. (2020). 'A semi-supervised approach for detection of scada attacks in gas pipeline control systems'. In: *2020 IEEE-HYDCON*. IEEE, pp. 1–8.

Junejo, Khurum Nazir and Jonathan Goh (2016). 'Behaviour-based attack detection and classification in cyber physical systems using machine learning'. In: *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, pp. 34–43.

Junejo, Khurum Nazir and David Yau (2016). 'Data driven physical modelling for intrusion detection in cyber physical systems'. In: *Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016*. IOS Press, pp. 43–57.

Kaplan, Jared et al. (2020). 'Scaling Laws for Neural Language Models'. In: *arXiv*. Shows predictable gains from more data/compute in deep models. eprint: 2001.08361. URL: https://arxiv.org/abs/2001.08361.

Khalili, Abdullah et al. (2015). 'SysDetect: a systematic approach to critical state determination for Industrial Intrusion Detection Systems using Apriori algorithm'. In: *Journal of Process Control* 32, pp. 154–160.

Kim, Bedeuro et al. (2023). 'A comparative study of time series anomaly detection models for industrial control systems'. In: *Sensors* 23.3, p. 1310.

Kim, Bongjun et al. (2023). 'A Comparative Study of Time Series Anomaly Detection for Industrial Control Systems'. In: *Sensors* 23.3, p. 9921147. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9921147/.

Kingma, Diederik P et al. (2014). 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980*.

Kiss, Istvan, Bela Genge et al. (2015). 'A clustering-based approach to detect cyber attacks in process control systems'. In: *2015 IEEE 13th international conference on industrial informatics (INDIN)*. IEEE, pp. 142–148.

Kiss, Istvan, Béla Genge et al. (2014). 'Data clustering-based anomaly detection in industrial control systems'. In: *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, pp. 275–281.

Kleinmann, Amit et al. (2018). 'Stealthy deception attacks against SCADA systems'. In: *Computer Security: ESORICS 2017 International Workshops, CyberICPS 2017 and SECPRE 2017, Oslo, Norway, September 14-15, 2017, Revised Selected Papers 3*. Springer, pp. 93–109.

Koc, Levent et al. (2012). 'A network intrusion detection system based on a Hidden Naıve Bayes multiclass classifier'. In: *Expert Systems with Applications* 39.18, pp. 13492–13500.

Kravchik, Moshe et al. (2018). 'Detecting cyber attacks in industrial control systems using convolutional neural networks'. In: *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pp. 72–83.

Krishnan, Ravikiran et al. (2015). 'Conditional distance based matching for one-shot gesture recognition'. In: *Pattern Recognition* 48.4, pp. 1302–1314.

Krotofil, Marina et al. (2015). 'The process matters: Ensuring data veracity in cyber-physical systems'. In: *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pp. 133–144.

Kumar, Vipin et al. (2013). 'K-means clustering approach to analyze NSL-KDD intrusion detection dataset'. In: *International Journal of Soft Computing and Engineering (IJSCE)* 3.4, pp. 1–4.

Kumara, Ajay et al. (2018). 'Automated multi-level malware detection system based on reconstructed semantic view of executables using machine learning techniques at VMM'. In: *Future Generation Computer Systems* 79, pp. 431–446.

Kung, Enoch et al. (2016). 'The Performance and Limitations of Stealthy Attacks on Higher Order Systems'. In: *IEEE Transactions on Automatic Control* 62.2, pp. 941–947.

Kurt, Mehmet Necip et al. (2018). 'Online cyber-attack detection in smart grid: A reinforcement learning approach'. In: *IEEE Transactions on Smart Grid* 10.5, pp. 5174–5185.

Kwon, YooJin et al. (2015). 'A behavior-based intrusion detection technique for smart grid infrastructure'. In: *2015 IEEE Eindhoven PowerTech*. IEEE, pp. 1–6.

Laan, Nathan et al. (2025). 'Implementing Cybersecurity for Industrial Connected Products: How to Use Standards to Protect Operational Technology'. In: *IEEE Industry Applications Magazine*.

Langner, Ralph (2011). 'Stuxnet: Dissecting a cyberwarfare weapon'. In: *IEEE Security & Privacy* 9.3, pp. 49–51.

Lapin, Maksim et al. (2014). 'Learning Using Privileged Information: SVM+ and Weighted SVM'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. URL: https://proceedings.neurips.cc/paper_files/paper/2014/hash/4180eac3031b,.

Larsson, Axel et al. (2025). *Introducing Homomorphic Encryption for Digital Twin Security: Enhancing Industrial Automation through Secure Data Processing*.

LeCun, Yann et al. (1998). 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Li, Dan et al. (2019). 'MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks'. In: *International conference on artificial neural networks*. Springer, pp. 703–716.

Li, Yu-Qi et al. (2020). 'AquaSee: Predict load and cooling system faults of supercomputers using chilled water data'. In: *Journal of Computer Science and Technology* 35, pp. 221–230.

Li, Xue et al. (2020). 'Robust learning with imperfect privileged information'. In: *Artificial Intelligence* 282, p. 103246.

Lin, Hui, Adam Slagell, Catello Di Martino et al. (2013). 'Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol'. In: *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, pp. 1–4.

Lin, Hui, Adam Slagell, Zbigniew Kalbarczyk et al. (2013). 'Semantic security analysis of SCADA networks to detect malicious control commands in power grids'. In: *Proceedings of the first ACM workshop on Smart energy grid security*, pp. 29–34.

Lin, Qin et al. (2018). 'TABOR: A graphical model-based approach for anomaly detection in industrial control systems'. In: *Proceedings of the 2018 on asia conference on computer and communications security*, pp. 525–536.

Linda, Ondrej, Milos Manic, Jim Alves-Foss et al. (2011). 'Towards resilient critical infrastructures: Application of Type-2 Fuzzy Logic in embedded network security cyber sensor'. In: *2011 4th international symposium on resilient control systems*. IEEE, pp. 26–32.

Linda, Ondrej, Milos Manic and Todd Vollmer (2012). 'Improving cyber-security of smart grid systems via anomaly detection and linguistic domain knowledge'. In: *2012 5th International Symposium on Resilient Control Systems*. IEEE, pp. 48–54.

Linda, Ondrej, Milos Manic, Todd Vollmer and Jason Wright (2011). 'Fuzzy logic based anomaly detection for embedded network security cyber sensor'. In: *2011 IEEE Symposium on computational intelligence in cyber security (CICS)*. IEEE, pp. 202–209.

Linda, Ondrej, Todd Vollmer et al. (2009). 'Neural network based intrusion detection system for critical infrastructures'. In: *2009 international joint conference on neural networks*. IEEE, pp. 1827–1834.

Liu, W. et al. (2014). 'Detecting Communities Based on Network Topology'. In: *Sci. Rep.* 4, p. 5739.

Liu, Yao et al. (2011). 'False data injection attacks against state estimation in electric power grids'. In: *ACM Transactions on Information and System Security (TISSEC)* 14.1, pp. 1–33.

Loo, Fu-Nie et al. (2023). 'Semi-supervised cyber-attack detection for industrial control system of water storage'. In: *System Innovation for a World in Transition*. CRC Press, pp. 70–76.

Lopez-Paz, David et al. (2015). 'Unifying distillation and privileged information'. In: *arXiv preprint arXiv:1511.03643*.

Loukas, George et al. (2017). 'Cloud-based cyber-physical intrusion detection for vehicles using deep learning'. In: *Ieee Access* 6, pp. 3491–3508.

Lu, Huimin et al. (2017). 'Motor anomaly detection for unmanned aerial vehicles using reinforcement learning'. In: *IEEE internet of things journal* 5.4, pp. 2315–2322.

Luo, Y (2013). 'Research and design on intrusion detection methods for industrial control system'. In: *Zhejiang University*.

Maglaras, Leandros A and Jianmin Jiang (2014). 'Intrusion detection in SCADA systems using machine learning techniques'. In: *2014 Science and Information Conference*. IEEE, pp. 626–631.

Maglaras, Leandros A, Jianmin Jiang and Tiago Cruz (2014). 'Integrated OCSVM mechanism for intrusion detection in SCADA systems'. In: *Electronics Letters* 50.25, pp. 1935–1936.

Makris, Antonios et al. (2025). 'CoEvolution: A Comprehensive Trustworthy Framework For Connected Machine Learning And Secure Interconnected AI Solutions'. In: *2025 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, pp. 838–845.

Marsden, Thomas et al. (2018). 'Probability risk identification based intrusion detection system for SCADA systems'. In: *Mobile Networks and Management: 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings 9*. Springer, pp. 353–363.

Mathur, Aditya P et al. (2016a). 'SWaT: A water treatment testbed for research and training on ICS security'. In: *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, pp. 31–36.

— (2016b). *SWaT: A Water Treatment Testbed for Research and Training on ICS Security*. URL: https://research.scy-phy.net/swat.pdf.

Maurya, Sudhanshu et al. (2024). 'Quantum Cryptography for Secure Communications in Industrial Mechatronics and Embedded Systems'. In: *2024 20th*

IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA). IEEE, pp. 1–8.

Mehmood, Merwa et al. (2024). 'The role of rule mining in generating synthetic cyber-physical system attack samples'. In: *IEEE Internet of Things Magazine* 7.6, pp. 14–19.

Mesbah, Mohamed et al. (2019). 'Cyber threats and policies for industrial control systems'. In: *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, pp. 1–6.

Mitchell, Robert et al. (2014). 'A survey of intrusion detection techniques for cyber-physical systems'. In: *ACM Computing Surveys (CSUR)* 46.4, pp. 1–29.

MITRE Corporation (2025a). *Brute Force I/O (T0806) — ICS — MITRE ATT&CK*. URL: https://attack.mitre.org/techniques/T0806/ (visited on 06/09/2025).

— (2025b). *Collection, Tactic TA0100 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0100/ (visited on 06/09/2025).

— (2025c). *Command and Control, Tactic TA0101 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0101/ (visited on 06/09/2025).

— (2025d). *Discovery, Tactic TA0102 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0102/ (visited on 06/09/2025).

— (2025e). *Evasion, Tactic TA0103 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0103/ (visited on 06/09/2025).

— (2025f). *Execution, Tactic TA0104 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0104/ (visited on 06/09/2025).

MITRE Corporation (2025g). *Impact, Tactic TA0105 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0105/ (visited on 06/09/2025).

— (2025h). *Impair Process Control, Tactic TA0106 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0106/ (visited on 06/09/2025).

— (2025i). *Inhibit Response Function, Tactic TA0107 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0107/ (visited on 06/09/2025).

— (2025j). *Initial Access, Tactic TA0108 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0108/ (visited on 06/09/2025).

— (2025k). *Lateral Movement, Tactic TA0109 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0109/ (visited on 06/09/2025).

— (2025l). *Persistence, Tactic TA0110 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0110/ (visited on 06/09/2025).

— (2025m). *Privilege Escalation, Tactic TA0111 — ICS — MITRE ATT&CK*. Live knowledge base. URL: https://attack.mitre.org/tactics/TA0111/ (visited on 06/09/2025).

— (2025n). *Unauthorized Command Message (T0855) — ICS — MITRE ATT&CK*. URL: https://attack.mitre.org/techniques/T0855/ (visited on 06/09/2025).

Modbus, IDA (2004). 'Modbus application protocol specification v1. 1a'. In: *North Grafton, Massachusetts (www. modbus. org/specs. php)*.

Momeni, Ahmadreza et al. (2018). 'Understanding LUPI (learning using privileged information)'. In: *Ionosphere* 201.7, p. 6.

Morris, Thomas and Wei Gao (2014). *Industrial Control System Cyber Attack Datasets (UAH site)*. URL: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets (visited on 07/09/2025).

Morris, Thomas, Rayford Vaughn et al. (2012). 'A retrofit network intrusion detection system for MODBUS RTU and ASCII industrial control systems'. In: *2012 45th Hawaii International Conference on System Sciences*. IEEE, pp. 2338–2345.

Moustafa, Nour et al. (2018). 'An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things'. In: *IEEE Internet of Things Journal* 6.3, pp. 4815–4830.

Moya, Christian et al. (2018). 'Application of correlation indices on intrusion detection systems: Protecting the power grid against coordinated attacks'. In: *arXiv preprint arXiv:1806.03544*.

Muda, Z et al. (2011). 'Intrusion detection based on K-Means clustering and Naıve Bayes classification'. In: *2011 7th international conference on information technology in Asia*. IEEE, pp. 1–6.

Mujeeb Ahmed, Chuadhry et al. (2021). 'Machine learning for cps security: applications, challenges and recommendations'. In: *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, pp. 397–421.

Murphy, Kevin P. (2022). *Probabilistic Machine Learning: An Introduction*. Covers i.i.d. and exchangeability assumptions in supervised learning. Cambridge, MA: MIT Press. ISBN: 978-0-262-04682-4. URL: https://probml.github.io/pml-book/book1.html.

Myers, David et al. (2017). 'Process discovery for industrial control system cyber attack detection'. In: *ICT Systems Security and Privacy Protection: 32nd IFIP TC 11 International Conference, SEC 2017, Rome, Italy, May 29-31, 2017, Proceedings 32*. Springer, pp. 61–75.

Nader, Patric et al. (2014a). 'Lp-norms in one-class classification for intrusion detection in SCADA systems'. In: *IEEE Transactions on Industrial Informatics* 10.4, pp. 2308–2317.

— (2014b). 'Mahalanobis-based one-class classification'. In: *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6.

Nankya, Mary et al. (2023). 'Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies'. In: *Sensors* 23.21, p. 8840.

Ndonda, Gorby Kabasele et al. (2019). 'A Public Network Trace of a Control and Automation System'. In: *arXiv*. eprint: 1908.02118. URL: https://arxiv.org/abs/1908.02118.

Nie, Zhijie et al. (2024). 'Intrusion Detection in Cyber-Physical Grid using Incremental ML with Adaptive Moment Estimation'. In: *IEEE Transactions on Industrial Cyber-Physical Systems*.

Otoum, Safa et al. (2017). 'Mitigating False Negative intruder decisions in WSN-based Smart Grid monitoring'. In: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 153–158. DOI: 10.1109/IWCMC.2017.7986278.

Oyedotun, Samuel Abiodun et al. (2025). 'Towards Intelligent Cybersecurity in SCADA and DCS Environments: Anomaly Detection Using Multimodal Deep Learning and Explainable AI'. In: *Journal of Science Research and Reviews* 2.3, pp. 20–31.

Pal, Koyena et al. (2017). 'Effectiveness of association rules mining for invariants generation in cyber-physical systems'. In: *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, pp. 124–127.

Palácios, José M Balbuena et al. (2013). 'Intrusion detection system: A hybrid approach for cyber-physical environments'. In: *Technology* 39, pp. 193–204.

Pan, Sinno Jialin et al. (2009). 'A survey on transfer learning'. In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Panfili, Martina et al. (2018). 'A game-theoretical approach to cyber-security of critical infrastructures based on multi-agent reinforcement learning'. In: *2018 26th Mediterranean Conference on Control and Automation (MED)*. IEEE, pp. 460–465.

Patel, Ahmed et al. (2017). 'A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems'. In: *Computers & Security* 64, pp. 92–109.

Patton, Ron J (1995). 'Robustness in model-based fault diagnosis: the 1995 situation'. In: *IFAC Proceedings Volumes* 28.12, pp. 45–67.

Paxson, Vern (1999). 'Bro: a system for detecting network intruders in real-time'. In: *Computer networks* 31.23-24, pp. 2435–2463.

Perales Gómez, Ángel Luis et al. (2020). 'Madics: A methodology for anomaly detection in industrial control systems'. In: *Symmetry* 12.10, p. 1583.

— (2023). 'An interpretable semi-supervised system for detecting cyberattacks using anomaly detection in industrial scenarios'. In: *IET Information Security* 17.4, pp. 553–566.

Ponomarev, Stanislav et al. (2015). 'Industrial control system network intrusion detection by telemetry analysis'. In: *IEEE Transactions on Dependable and Secure Computing* 13.2, pp. 252–260.

Pordelkhaki, Moojan et al. (2021). 'Intrusion Detection for Industrial Control Systems by Machine Learning using Privileged Information'. In: *2021 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, pp. 1–6.

Prasanna, SS et al. (2023). 'Anomaly-based Intrusion Detection System for ICS'. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, pp. 1–4.

Qi, Ruobin et al. (2021). 'Detecting cyber attacks in smart grids using semi-supervised anomaly detection and deep representation learning'. In: *Information* 12.8, p. 328.

Qiao, Ziyan et al. (2024). 'Reinforcement Learning-based Security Enhancement for Controlled Optimization of Phases in Intelligent Traffic Signal System'. In: *IEEE Transactions on Industrial Cyber-Physical Systems*.

Qin, Qiang et al. (2026). 'Robot digital twin systems in manufacturing: Technologies, applications, trends and challenges'. In: *Robotics and Computer-Integrated Manufacturing* 97, p. 103103.

Raman, MR Gauthama et al. (2020). 'Deep autoencoders as anomaly detectors: Method and case study in a distributed water treatment plant'. In: *Computers & Security* 99, p. 102055.

Ren, Wenyu et al. (2018). 'Edmand: Edge-based multi-level anomaly detection for scada networks'. In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, pp. 1–7.

Roesch, Martin et al. (1999). 'Snort: Lightweight intrusion detection for networks.' In: *Lisa*. Vol. 99. 1, pp. 229–238.

Romera-Paredes, Bernardino et al. (2015). 'An embarrassingly simple approach to zero-shot learning'. In: *International conference on machine learning*. PMLR, pp. 2152–2161.

Ruan, Jiaqi et al. (2023). 'Deep learning for cybersecurity in smart grids: Review and perspectives'. In: *Energy Conversion and Economics* 4.4, pp. 233–251.

Ruslan, Kassym et al. (2025). 'Clustering Protocols and Algorithms for IoT and IIoT Using Artificial Intelligence: Review, Classifications and Challenges'. In: *2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)*, pp. 1430–1436. DOI: 10.1109/AIITA65135.2025.11048004.

Sahu, Shailendra et al. (2015). 'Network intrusion detection system using J48 Decision Tree'. In: *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, pp. 2023–2026.

Samara, Ghassan et al. (2024). 'Deep Learning Approaches for Malware Detection in the Industrial Internet of Things: A Comprehensive Analysis'. In: *2024 25th International Arab Conference on Information Technology (ACIT)*. IEEE, pp. 1–7.

Samiah, Abdul et al. (2025). 'Decision tree based invariants for intrusion detection in industrial control system'. In: *Computers & Security*, p. 104511.

Sangoleye, Fisayo et al. (2024). 'Intrusion Detection in Industrial Control Systems based on Deep Reinforcement Learning'. In: *IEEE Access*.

Sapello, Angelo et al. (2017). 'Application of learning using privileged information (LUPI): botnet detection'. In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, pp. 1–8.

Scarfone, Karen et al. (2007). 'Guide to intrusion detection and prevention systems (idps)'. In: *NIST special publication* 800.2007, p. 94.

Schneider, Peter et al. (2018). 'High-performance unsupervised anomaly detection for cyber-physical system networks'. In: *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pp. 1–12.

Shaikh, Tawseef Ayoub et al. (2020). 'Transfer learning privileged information fuels CAD diagnosis of breast cancer'. In: *Machine Vision and Applications* 31.1, p. 9.

Sharma, Richa et al. (2023). 'Detection of false data injection in smart grid using PCA based unsupervised learning'. In: *Electrical Engineering* 105.4, pp. 2383–2396.

Sharmanska, Viktoriia et al. (2013). 'Learning to rank using privileged information'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 825–832.

Shen, Lu et al. (2020). 'Single-modal neuroimaging computer aided diagnosis for schizophrenia based on ensemble learning using privileged information'. In: *Sheng wu yi xue Gong Cheng xue za zhi= Journal of Biomedical Engineering= Shengwu Yixue Gongchengxue Zazhi* 37.3, pp. 405–411.

Shen, Wuqiang et al. (2024). 'CluSAD: Self-Supervised Learning-Based Anomaly Detection for Industrial Control Systems'. In: *2024 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI)*. IEEE, pp. 545–552.

Shin, Hyeong-Keun et al. (2021). 'Two ICS Security Datasets and Anomaly Detection Contest on the HIL-based Augmented ICS Testbed'. In: *USENIX CSET*. URL: https://dl.acm.org/doi/10.1145/3474718.3474719.

Shinde, Pramila P et al. (2018). 'A review of machine learning and deep learning applications'. In: *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. IEEE, pp. 1–6.

Singh, Abhishek (2024). 'Enhancing Cybersecurity for Digital Twins: Challenges and Solutions'. In: *International Journal on Science and Technology* 14.4.

Singh, Suby et al. (2020). 'Artificial intelligence and security of industrial control systems'. In: *Handbook of Big Data Privacy*, pp. 121–164.

Slay, Jill et al. (2007). 'Lessons learned from the maroochy water breach'. In: *International conference on critical infrastructure protection*. Springer, pp. 73–82.

Slowik, Joseph (2019). 'Evolution of ICS attacks and the prospects for future disruptive events'. In: *Threat Intelligence Centre Dragos Inc* 162.

Socher, Richard et al. (2013). 'Zero-shot learning through cross-modal transfer'. In: *Advances in neural information processing systems* 26.

Sridhar, Siddharth et al. (2014). 'Model-based attack detection and mitigation for automatic generation control'. In: *IEEE Transactions on Smart Grid* 5.2, pp. 580–591.

Stockman, Melissa et al. (2019). 'Detecting control system misbehavior by fingerprinting programmable logic controller functionality'. In: *International Journal of Critical Infrastructure Protection* 26, p. 100306.

Stouffer, Keith et al. (2023). 'Guide to operational technology (ot) security'. In.

Suaboot, Jakapan et al. (2020). 'A Taxonomy of Supervised Learning for IDSs in SCADA Environments'. In: *ACM Computing Surveys (CSUR)* 53.2, pp. 1–37.

Tian, Jue et al. (2018). 'Enhanced hidden moving target defense in smart grids'. In: *IEEE transactions on smart grid* 10.2, pp. 2208–2223.

Tsai, Chih-Fong et al. (2009). 'Intrusion detection by machine learning: A review'. In: *expert systems with applications* 36.10, pp. 11994–12000.

Tsang, Chi-Ho et al. (2005). 'Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction'. In: *2005 IEEE international conference on industrial technology*. IEEE, pp. 51–56.

Tsvetanov, Tsvetan et al. (2021). 'The effect of the Colonial Pipeline shutdown on gasoline prices'. In: *Economics Letters* 209, p. 110122.

Turnipseed, Ian P. (2015). 'A New SCADA Dataset for Intrusion Detection System Research'. PhD thesis. Mississippi State University. URL: https://scholarsjunction.msstate.edu/cgi/viewcontent.cgi?article=1208&context=td.

Turrin, Federico et al. (2020). *A Statistical Analysis Framework for ICS Process Datasets (WADI description)*. URL: https://publications.cispa.saarland/3279/1/Analysis_on_water_distribution_system_datasets.pdf (visited on 07/09/2025).

Umer, Muhammad Azmi et al. (2017). 'Integrating design and data centric approaches to generate invariants for distributed attack detection'. In: *Proceedings of the 2017 workshop on cyber-physical systems security and privacy*, pp. 131–136.

— (2020). 'Generating invariants using design and data-centric approaches for distributed attack detection'. In: *International Journal of Critical Infrastructure Protection* 28, p. 100341.

Urbina, David I et al. (2016). 'Limiting the impact of stealthy attacks on industrial control systems'. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 1092–1105.

Urrea, Claudio (2025). 'Artificial Intelligence-Driven and Bio-Inspired Control Strategies for Industrial Robotics: A Systematic Review of Trends, Challenges, and Sustainable Innovations Toward Industry 5.0'. In: *Machines* 13.8, p. 666.

V, Gayathry et al. (2025). 'Assessing the Effectiveness of Machine Learning in Cyber Attack Mitigation: A Review'. In: *2025 8th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 762–771. DOI: 10.1109/ICOEI65986.2025.11013517.

Valdes, Alfonso et al. (2009). 'Communication pattern anomaly detection in process control systems'. In: *2009 IEEE Conference on Technologies for Homeland Security*. IEEE, pp. 22–29.

Vapnik, Vladimir (2006). *Estimation of dependences based on empirical data*. Springer Science & Business Media.

Vapnik, Vladimir and Akshay Vashist (2009a). 'A New Learning Paradigm: Learning Using Privileged Information'. In: *Neural Networks* 22.5-6, pp. 544–557. DOI: 10.1016/j.neunet.2009.06.042.

— (2009b). 'A new learning paradigm: Learning using privileged information'. In: *Neural networks* 22.5-6, pp. 544–557.

Varghese, Josy Elsa et al. (2024). 'A Survey DDoS Attack in IoT Cloud Platform'. In: *2024 Eighth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 388–393. DOI: 10.1109/PDGC64653.2024.10984361.

Vollmer, Todd et al. (2009). 'Computationally efficient neural network intrusion security awareness'. In: *2009 2nd International Symposium on Resilient Control Systems*. IEEE, pp. 25–30.

Williams, Theodore J (1994). 'The Purdue enterprise reference architecture'. In: *Computers in industry* 24.2-3, pp. 141–158.

Wu, Di et al. (2012). 'One shot learning gesture recognition from rgbd images'. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, pp. 7–12.

Xiao, Liyuan et al. (2014). 'Bayesian model averaging of Bayesian network classifiers for intrusion detection'. In: *2014 IEEE 38th International Computer Software and Applications Conference Workshops*. IEEE, pp. 128–133.

Xu, Xinxing et al. (2015). 'Distance metric learning using privileged information for face verification and person re-identification'. In: *IEEE transactions on neural networks and learning systems* 26.12, pp. 3150–3162.

Yan, Jun et al. (2016). 'Q-learning-based vulnerability analysis of smart grid against sequential topology attacks'. In: *IEEE Transactions on Information Forensics and Security* 12.1, pp. 200–210.

Yan, Siwen et al. (2023). 'Learning with privileged and sensitive information: a gradient-boosting approach'. In: *Frontiers in artificial intelligence* 6, p. 1260583.

Yang, Dayu et al. (2006). 'Anomaly-based intrusion detection for SCADA systems'. In: *5th intl. topical meeting on nuclear plant instrumentation, control and human machine interface technologies (npic&hmit 05)*. Citeseer, pp. 12–16.

Yang, Shuo et al. (2022). 'Toward understanding privileged features distillation in learning-to-rank'. In: *Advances in Neural Information Processing Systems* 35, pp. 26658–26670.

Yang, Xun et al. (2017). 'Person re-identification with metric learning using privileged information'. In: *IEEE Transactions on Image Processing* 27.2, pp. 791–805.

Yang, Yi et al. (2013). 'Rule-based intrusion detection system for SCADA networks'. In.

Yau, Ken et al. (2017). 'Detecting anomalous behavior of PLC using semi-supervised machine learning'. In: *2017 IEEE Conference on Communications and Network Security (CNS)*. IEEE, pp. 580–585.

Yin, Xiaoxin et al. (2006). 'Efficient classification across multiple database relations: A crossmine approach'. In: *IEEE Transactions on Knowledge and Data Engineering* 18.6, pp. 770–783.

Yusheng, Wang et al. (2017). 'Intrusion detection of industrial control system based on Modbus TCP protocol'. In: *2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)*. IEEE, pp. 156–162.

Zemanek, Sven et al. (2022). 'PowerDuck: A GOOSE Data Set of Cyberattacks in Substations'. In: *arXiv*. eprint: 2207.04716. URL: https://arxiv.org/abs/2207.04716.

Zhang, Qi et al. (2017). 'A fuzzy probability Bayesian network approach for dynamic cybersecurity risk assessment in industrial control systems'. In: *IEEE Transactions on Industrial Informatics* 14.6, pp. 2497–2506.

Zhang, Yichi et al. (2011). 'Distributed intrusion detection system in a multi-layer network architecture of smart grids'. In: *IEEE Transactions on Smart Grid* 2.4, pp. 796–808.

Zhou, Chunjie et al. (2020). 'A unified architectural approach for cyberattack-resilient industrial control systems'. In: *Proceedings of the IEEE* 109.4, pp. 517–541.

Zizzo, Giulio et al. (2019). 'Adversarial machine learning beyond the image domain'. In: *Proceedings of the 56th Annual Design Automation Conference 2019*, pp. 1–4.