AZERBAIJAN STATE OIL AND INDUSTRY UNIVERSITY
(ASOIU)
BA programs (MBA, BBA, ZU)

Master of Business Administration Program

AI-DRIVEN THREAT DETECTION IN INDUSTRIAL CONTROL SYSTEMS SECURITY

Major:	**Computer Information System**
Student:	**Rahimli Sanan**
Supervisor:	**DtS. Zeynalova Lala**
Program director:	**Professor Rafik Aliyev**

BAKU – 2025

ABSTRACT

AI-DRIVEN THREAT DETECTION IN INDUSTRIAL CONTROL SYSTEMS SECURITY

This dissertation addresses the urgent need for robust intrusion detection in Industrial Control Systems (ICS) underpinning electric transmission networks. Traditional solutions focusing solely on network traffic or process measurements suffer from high false-negative and false-positive rates, leaving critical infrastructure vulnerable to cyber-physical attacks that can mimic legitimate disturbances. To overcome these limitations, we developed a comprehensive, multi-modal framework that integrates four data sources synchrophasor measurements, network IDS logs, control-panel events, and relay-status signals across 37 realistic event scenarios.

A high-fidelity SCADA dataset was synthesized to support binary (normal vs. attack), ternary (normal vs. fault vs. attack), and detailed multi-class (37 event types) classification. Feature engineering combined statistical preprocessing with Gini-impurity–based selection, yielding a focused subset of high-value attributes. Three model families were implemented and compared: traditional machine-learning classifiers (Random Forests, Support Vector Machines), deep-learning architectures (LSTM autoencoders, CNN-LSTM hybrids), and a novel hybrid pipeline that pairs unsupervised anomaly screening with supervised signature validation.

Performance was evaluated using precision, recall, F1-score, ROC/AUC, and time-to-detect metrics under cost-sensitive thresholds prioritizing false-negative reduction. Random Forests achieved F1-scores above 0.93 in binary tasks with low latency, while CNN-LSTM models excelled in multi-class detection (mean AUC 0.97). The hybrid approach reduced false positives by 25% while maintaining recall above 95%. Feature-importance analysis identified phase-angle deviations and relay-log patterns as key indicators, guiding optimal sensor placement.

These findings demonstrate that a layered, data-driven detection strategy can significantly enhance ICS security without compromising real-time performance. The dissertation contributes a publicly extensible dataset, a validated hybrid detection architecture, and actionable insights for deploying AI-driven defenses in SCADA environments, thereby advancing the resilience of critical infrastructure against evolving cyber-physical threats.

Keywords: Industrial Control Systems; SCADA Security; Intrusion Detection Systems; Machine Learning; Multi-modal Data Integration.

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr. Lala Zeynalova, for their expert guidance in artificial intelligence, throughout the simulation and implementation and for their invaluable feedback throughout this research. Their insight and encouragement were instrumental in shaping both the conceptual framework and technical execution of this work.

Furthermore, I extend my thanks to the experts who contributed to the validation survey for this research project.

DECLARATION

I hereby declare that this dissertation titled “AI-Driven Threat Detection in Industrial Control Systems Security” is my own original work and has not been submitted in any form for any degree or diploma at any other university. All sources of information and data have been duly acknowledged and referenced. Any contributions from others have been explicitly indicated.

Rahimli Sanan

TABLE OF CONTENTS

.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SCADA - Supervisory Control And Data Acquisition

ANN - Artificial Neural Network

CNN - Convolutional Neural Network

RNN - Recurrent Neural Network

GAN - Generative Adversarial Nets

AUC - Area Under the (ROC) Curve

CVE - Common Vulnerabilities and Exposures

DMZ - Demilitarized Zone

GIWRF - Gini Impurity–based Weighted Random Forest

HART - Highway Addressable Remote Transducer

HMI - Human–Machine Interface

IDS - Intrusion Detection System

IIoT - Industrial Internet of Things

LSTM - Long Short-Term Memory (network)

MES - Manufacturing Execution System

NIST - National Institute of Standards and Technology

OPC UA - OPC Unified Architecture

PLC - Programmable Logic Controller

RTU - Remote Terminal Unit

SIEM - Security Information and Event Management

SIL - Safety Integrity Level

SWaT - Secure Water Treatment (testbed/dataset)

TCN - Temporal Convolutional Network

VFD - Variable-Frequency Drive

DNP3 - Distributed Network Protocol version 3

DP - Deterministic Performance

GRU - Gated Recurrent Unit

INTRODUCTION

Industrial Control Systems (ICS) underpin critical infrastructure sectors such as energy, water treatment, and manufacturing by coordinating physical processes through a combination of hardware and software components. As these systems become increasingly interconnected and digitized, they face evolving cyber threats that exploit legacy vulnerabilities and complex network architectures. This literature review investigates the state of ICS security, with a focus on machine learning–driven defenses and emerging detection techniques.

Statement of Problem: Despite extensive research on ICS security architectures and threat mitigation, operators still struggle to deploy robust, real-time intrusion detection tailored to the unique constraints of industrial environments. Legacy control devices often lack built-in cybersecurity measures, and conventional IT-based defenses can introduce latency or compatibility issues in time-sensitive processes. This review seeks to clarify:

What are the prevailing gaps in ICS intrusion detection and defense architectures that hinder reliable protection of critical processes?

Purpose of the Study: The purpose of this study is to synthesize existing research on ICS security covering architectural controls, historical incidents, and machine learning–based defense strategies to identify best practices and outstanding challenges. By systematically reviewing the literature, the study aims to:

- Highlight how machine learning and anomaly detection techniques have been applied to ICS environments.
- Uncover limitations in current approaches regarding accuracy, scalability, and real-time deployment.

- Lay the groundwork for proposing an integrated detection framework in subsequent research.

Significance of the Study: Securing ICS is vital for maintaining national and economic security, as successful cyberattacks can disrupt essential services and cause physical damage. High-profile incidents such as Stuxnet and the Colonial Pipeline attack demonstrate both the potential impact and the urgency of developing advanced detection mechanisms that balance safety with operational performance. This review will inform researchers and practitioners about the most effective defense-in-depth strategies and machine learning applications, justifying further investment in AI-driven security solutions.

Research Questions

1. Which architectural controls and layered defense strategies have proven most effective against ICS cyber threats?
2. How have machine learning and anomaly detection models been tailored for ICS environments and what are their performance trade-offs?
3. What datasets and benchmarking practices exist for evaluating ICS intrusion detection systems?
4. Where do current methodologies fall short, and what research opportunities remain for integrating novel AI techniques?

Review of the Literature: Prior studies have examined ICS architectures and protocols, documented notable security breaches and surveyed machine learning approaches for intrusion detection. Research on time-series anomaly detection, clustering methods, and deep learning models reveals a rich landscape of techniques yet integration into real-world ICS remains limited. This section synthesizes these contributions, grouping them by theme to demonstrate both progress and gaps.

Research Methodology: This review employs a systematic literature synthesis. Relevant papers were identified from digital libraries and the provided reference list, focusing on works published between 200 6 and 2025. Each article was evaluated for:

- Scope (architectural vs. algorithmic focus)
- Methodology (empirical case study, simulation, theoretical analysis)
- Key Findings (detection performance, deployment challenges) The synthesized findings inform the discussion of best practices and research directions.

Delimitations, Limitations, and Assumptions: Only peer-reviewed English-language publications. Coverage limited to ICS/SCADA environments, excluding broader IT networks. Potential omission of industry white papers and unpublished technical reports. Variability in experimental setups may hinder direct comparability of detection metrics. Reported results in selected studies are accurate and reproducible. Defined ICS architectures in the literature reasonably represent real-world deployments.

Research Novelty: While many reviews cover either architectural controls or machine learning models in isolation, this study uniquely bridges both domains proposing a framework that integrates layered defense principles with adaptive, AI-driven anomaly detection, tailored to the timing and reliability constraints of ICS.

Structure of the Thesis

This research study consists of the following parts:

- **Chapter 1:** Theoretical analysis of the AI-driven threat detection in industrial control systems security
- **Chapter 2:** Applications of artificial intelligence in industrial control systems security and threat detection
- **Chapter 3:** Simulation, implementation and results

CHAPTER 1. THEORETICAL ANALYSIS OF THE AI-DRIVEN THREAT DETECTION IN INDUSTRIAL CONTROL SYSTEMS SECURITY

1.1 ANALYSIS AND OUTLINING OF THE CORE ASPECTS OF THE AI-DRIVEN THREAT DETECTION

As cyber threats become increasingly sophisticated and targeted, the security of Industrial Control Systems (ICS) has emerged as a paramount concern. Existing research frequently points out the shortcomings of conventional intrusion detection systems, especially within dynamic, real-time industrial environments. This section offers a comprehensive overview of significant advances in AI-driven intrusion detection, underscoring how machine learning, data fusion, and hybrid detection methodologies are bolstering ICS security.

Machine learning is being recognized as a promising avenue to address the escalating cyber threats against Industrial Control Systems, particularly as the pace of digital transformation accelerates under Industry 4.0. This detailed review examines the inherent challenges faced by traditional intrusion detection systems and maps the current vulnerability landscape of modern ICS. It further assesses various machine learning-based techniques, highlighting their capacity to improve anomaly detection, threat classification, and real-time response capabilities within critical infrastructure [1].

A systematic review on anomaly detection techniques specifically designed for Industrial Control Systems is presented, covering both theoretical frameworks and practical applications. The analysis encompasses supervised, unsupervised, and semi-supervised learning methods employed in ICS cybersecurity, emphasizing their effectiveness in identifying unusual activities across industrial networks. Critical issues such as data imbalance, the scarcity of labeled datasets, and the necessity for real-time detection are also addressed, positioning this work as a vital resource for advancing threat detection strategies in ICS settings [2].

The usefulness of deep learning techniques as potent instruments for cybersecurity in industrial control systems is highlighted by this source. It explores how to identify cyber abnormalities using autoencoders, recurrent neural networks (RNNs), and convolutional neural networks (CNNs). Particularly in the high-stakes, real-time operational contexts of industrial systems, the significance of high-quality datasets, effective training, and model interpretability is emphasized. In order to fully utilize deep learning in ICS security, the article also identifies current research gaps that must be filled [3].

An exploration of ensemble learning techniques applied to ICS security demonstrates their superior capacity to manage intricate classification challenges and enhance detection performance. The research illustrates how the integration of multiple models can decrease false positives and improve the resilience of anomaly detection systems. The study includes comparative performance evaluations of widely used ensemble methods like bagging, boosting, and stacking in various ICS threat scenarios, affirming their practical utility for deployment in vital infrastructures [4].

With the growing digitalization and interconnectedness of industrial environments, the cybersecurity of Industrial Control Systems (ICS) has become a critical imperative. Traditional defensive measures frequently prove inadequate in countering the sophisticated and dynamic nature of contemporary threats. In this context, recent studies highlight the limitations of conventional detection approaches and explore how machine learning techniques are being leveraged to strengthen ICS defenses. One such investigation specifically examines the expanding threat landscape driven by Industry 4.0 and reviews machine learning advancements aimed at improving threat detection and resilience in ICS environments [5].

In a variety of industries, including manufacturing, utilities, transportation, and energy, industrial control systems (ICS) are critical to the automation and management of critical industrial operations. In order to monitor and regulate physical operations and guarantee efficiency and safety, these systems combine hardware, software, and communication networks. The advent of Industry 4.0 technologies and the growing integration of ICS with corporate IT networks, however, have made these formerly isolated systems more vulnerable to a variety of cybersecurity threats. This connectivity creates vulnerabilities stemming from legacy architectures, insufficient security controls, and advanced cyberattack techniques. Consequently, safeguarding ICS environments is a major concern for organizations and governments globally, given the potential for severe operational disruptions, safety hazards, and economic losses. This overview aims to detail the core components and communication protocols of ICS, identify common security challenges, and investigate emerging defense mechanisms, including the use of machine learning, to enhance the resilience and security posture of these critical systems [6].

It is suggested that Industrial Control Systems (ICS) networks use a defense-in-depth cybersecurity framework based on the Purdue Enterprise Reference Architecture (PERA) paradigm. This work systematically identifies and evaluates security deficiencies across various industrial domains through an extensive literature review, encompassing books, articles, and conference proceedings. It presents a conceptual structure designed to create a secure and resilient operational environment for ICS and Operational Technology (OT) networks. This methodology underscores the crucial need for continuous protection and uninterrupted operation within these infrastructures, offering a structured cybersecurity architecture uniquely suited to industrial systems [7].

Cyber-Physical Systems (CPS) integrate computation, networking, and physical processes through continuous feedback loops. Nevertheless, current discrete and virtualized computing models struggle to accurately represent the continuous and time-sensitive characteristics of these systems. While real-time operating systems, specialized hardware, middleware, and networking solutions offer partial improvements, a fundamental gap in abstraction remains. The author advocates for the development of a new foundational systems science that unifies physical and computational models, aiming to bridge this gap and more effectively address the distinct requirements of CPS [8].

Supervisory Control and Data Acquisition (SCADA) systems provide real-time monitoring and control for vital infrastructure, including power generation, oil extraction, and water treatment facilities. This study scrutinizes the essential hardware and software elements of SCADA, such as Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs), and Human-Machine Interfaces (HMIs). It further explores SCADA's network architecture and primary communication protocols like Modbus, DNP3, and IEC 60870. The review concludes by evaluating current security practices and outlining future strategies to fortify the resilience of SCADA networks [9].

This study investigates the role of SCADA systems in meeting the growing demands for reliability, security, and efficiency in power generation. It achieves this by first reviewing the functional and security prerequisites for deploying SCADA in power plants, then validating these requirements through the creation of a small-scale hydroelectric control and supervision prototype using Kentima AB's WideQuick software [10].

SCADA (Supervisory Control and Data Acquisition) is a software-driven process control system that interfaces with field devices such as Programmable Logic Controllers (PLCs) to monitor and manage operations distributed across geographical areas. It integrates computers, controllers, sensors, actuators, networks, and user interfaces to collect and process data in real time. SCADA systems are broadly applied in heavy industries like steel manufacturing, power generation, chemical processing, and experimental research, supporting thousands to hundreds of thousands of input/output channels [11].

Real-time responsiveness is paramount for SCADA systems that oversee critical infrastructure. This study compares SCADA middleware running on standard Windows desktop platforms versus Windows CE (Compact Embedded) platforms to ascertain which better supports time-sensitive operations. Through specific performance tests and analysis, the research demonstrates that Windows CE offers significantly improved predictability and responsiveness, rendering it more suitable for SCADA applications requiring real-time capabilities. These findings validate the preference for Windows CE over desktop operating systems in scenarios demanding stringent timing and reliability [12].

Integrating heterogeneous industrial control systems necessitates standardized communication frameworks. This paper proposes an OPC (OLE for Process Control)-based extension to centralize data exchange in SCADA systems through a unified API layer. This approach enables real-time monitoring and identification of a second-order industrial process, simultaneously logging data to both TDMS files and a MySQL database. Furthermore, an embedded web server provides live data visualization and remote access via intranet or internet connections, facilitating centralized supervision and remote management of industrial processes [13].

Modern microgrids require intelligent supervisory systems that effectively coordinate distributed components. This research designs and implements a Java-based SCADA middleware, connecting a microgrid's lower-level central controller with an upper-level web monitoring platform. Key functionalities such as real-time data acquisition, load balancing, secure concurrent processing, and control instruction parsing are validated in a live microgrid setting. The outcomes demonstrate enhancements in system reliability, operational safety, and economic performance, highlighting middleware's role as the intelligent core in microgrid management [14].

With the ascent of IoT and digital transformation, traditional SCADA architectures necessitate re-evaluation for improved performance. This study conducts an extensive review of ten contemporary SCADA architectures utilizing a literature-based Multi-Criteria Decision-Making matrix to assess factors such as scalability, security, and efficiency. Based on this evaluation, an optimal IoT-enabled SCADA architecture tailored for a power utility is put forth. A comparative analysis with the existing system reveals substantial gains in operational efficiency, system scalability, and resilience, underscoring the advantages of IoT integration in industrial control [15].

Interoperability poses a significant challenge in evolving SCADA systems that must integrate diverse components. This research introduces an Interoperability Prediction Framework (IPF) designed to aid early-stage SCADA product architecture by facilitating the transition from monolithic to loosely coupled systems. Two architectural models are developed and validated using the IPF: one focuses on reducing coupling between SCADA and Energy Management Systems (EMS) to enhance internal interoperability; the other promotes the integration of external components to improve external interoperability. Results indicate improved interoperability and potential for reduced long-term system complexity, despite the need for considerable initial development effort [16].

The increasing integration of battery energy storage systems into power grids demands efficient control and monitoring solutions. To address this need, the study develops a cost-effective, IoT-based SCADA system specifically designed for remote control and supervision of grid-connected inverters. After assessing existing SCADA platforms, a core system is built that incorporates an automatic control algorithm optimizing inverter operation based on dynamic energy prices and renewable generation patterns. Testing confirms that the system fulfills operational requirements while boosting economic benefits through predictive and adaptive control [17].

Industry 4.0 is transforming industrial automation but also exposes critical infrastructures to evolving cyber threats. This article underscores the vital role of cybersecurity in safeguarding industrial automation systems within Industry 4.0 frameworks. Emphasis is placed on comprehensive risk management, adherence to international standards like ISA/IEC 62443, and the implementation of layered security architectures. The discussion covers essential practices including continuous network monitoring, workforce training, incident response planning, and the necessity of organizational collaboration and policy enforcement to adapt against dynamic cyber risks [18].

Accurate anomaly detection is crucial for protecting Industrial Control Systems (ICS) from cyber-attacks and operational failures. This study evaluates the performance of five prominent time series anomaly detection models InterFusion, RANSynCoder, GDN, LSTM-ED, and USAD using benchmark ICS datasets (SWaT and HAI). The comparative analysis reveals that no single model universally outperforms others: InterFusion achieves the highest F1-score on the SWaT dataset, while RANSynCoder performs best on HAI. Notably, the research indicates that utilizing only 40% of the training data can yield results comparable to full training, suggesting a path toward more resource-efficient ICS anomaly detection solutions [19].

The IEEE Spectrum article chronicles the origins and operations of the Stuxnet worm, identified as the first known cyberweapon engineered to sabotage industrial systems. It infiltrated Iranian nuclear facilities by exploiting Siemens SCADA software and Windows vulnerabilities, subsequently altering programmable logic controllers. The piece highlights the cyber-physical ramifications and geopolitical consequences of cyber warfare [20].

A Reuters article offers a factual summary of the Stuxnet malware, explaining its dissemination via USB devices and its specific targeting of Siemens SCADA systems. It emphasizes expert consensus that the worm's complexity suggests nation-state involvement, with Iran as the primary target, marking a pivotal moment in industrial cybersecurity [21].

Historical cyber incidents offer valuable insights into the vulnerabilities inherent in industrial control systems. This paper re-examines the notorious 2000 Maroochy Water Services incident in Australia, where a disgruntled insider exploited weaknesses in the sewage SCADA system. By manipulating control processes, the attacker caused significant environmental pollution and operational disruptions. This event exposed the inherent risks of open and internet-connected SCADA architectures and has since become a crucial case study driving advancements in SCADA security research, risk management practices, and policy development to prevent similar future breaches [22].

The May 2021 Colonial Pipeline ransomware attack, carried out by the DarkSide group, leveraged a compromised VPN password to access IT systems, resulting in significant fuel shortages and price surges across the

Eastern US. The incident exposed critical infrastructure vulnerabilities, prompting new TSA cybersecurity regulations for pipeline operators. Key takeaways include the need for proactive threat mitigation, robust employee training, effective incident response, and strong public-private partnerships to bolster infrastructure resilience against evolving cyber threats [23, 24].

Artificial Intelligence (AI) has undergone profound transformations from its foundational conceptual roots to its current ubiquitous presence across diverse industries. This paper reviews the historical progression of AI, tracing its beginnings from ancient philosophical concepts to its formal establishment in the 1950s. It highlights key milestones such as breakthroughs in symbolic logic, machine learning, and neural networks, particularly noting rapid progress in the 21st century driven by enhanced computing power and algorithms. The paper also emphasizes AI's extensive influence across fields like healthcare, finance, and robotics, demonstrating its crucial role in shaping modern technology and society [25].

Accurate statistical analysis is indispensable in healthcare research for extracting meaningful insights. This article provides a comprehensive review of linear regression analysis, focusing on its widespread application in vision science and medical studies. It covers both simple and multiple linear regression models, stressing the importance of correctly interpreting regression coefficients, validating assumptions, and selecting appropriate variables. The paper offers a practical checklist for editors and reviewers to evaluate the quality of regression-based studies and encourages collaboration with expert statisticians to enhance the reliability of research findings [26].

When researchers analyze categorical outcomes, especially binary ones, logistic regression proves to be an essential tool. This article explains the principles and application of logistic regression, distinguishing it from multiple linear regression by its capacity to handle binomial response variables. It illustrates how logistic regression calculates odds ratios to assess the effect of multiple explanatory variables while controlling for confounders. Through practical examples, the paper simplifies the interpretation of logistic regression results and discusses critical considerations for accurate use and reporting in research [27].

By combining several models, ensemble approaches in machine learning increase prediction accuracy. This article introduces Random Forests, a potent technique developed by Leo Breiman that extends bagging approaches to improve classification and regression tasks. Random Forest models can process both categorical and continuous predictor variables and outcomes, making them adaptable tools in various predictive scenarios. The paper explains how Random Forests balance bias and variance, presenting robust alternatives to boosting methods widely used in data science [28].

Understanding the varied nature of Micro, Small, and Medium Enterprises (MSMEs) is crucial for informed economic planning. This study employs clustering techniques, specifically DBSCAN and K-Means, to categorize MSMEs based on asset values and turnover, revealing distinct patterns in their characteristics and geographical distribution. By comparing the strengths of DBSCAN's noise management with K-Means' proximity grouping, the research offers complementary perspectives on MSME segmentation. The results offer practical advice for legislators hoping to promote MSME expansion and point the way toward directions for further research that includes more factors for a more thorough examination [29].

This survey delivers a structured and comprehensive overview of anomaly detection techniques across various research domains and applications. It classifies existing methods based on their underlying approaches and highlights the key assumptions each technique uses to differentiate normal from anomalous behavior. For each category, a foundational technique is presented along with its variations, offering a clear framework for understanding. The survey discusses the advantages, disadvantages, and computational complexity of these techniques, aiding in assessing their effectiveness across different domains. Overall, its aim is to deepen understanding of anomaly detection research and encourage cross-domain application of techniques [30].

Anomaly detection in time series data is a critical task for maintaining operational integrity and security across numerous sectors, from finance and healthcare to cybersecurity and industrial control systems. This thesis addresses the complexities involved in identifying anomalies by first stressing the need to thoroughly grasp the characteristics of time series data and the nature of anomalies before selecting a method. It provides a practical framework guiding researchers and practitioners through the process of investigating, selecting, and empirically testing various time series anomaly detection (TSAD) algorithms. By systematically analyzing performance results, the work seeks to deliver clear, actionable recommendations that balance cutting-edge innovation with well-established, explainable detection methodologies, ultimately improving anomaly detection efficacy in diverse applications [31, 32].

Reinforcement learning has achieved significant advancements with the advent of Deep Q-Networks (DQNs), which integrate the power of deep neural networks with decision-making in intricate environments. This paper reviews the progression of DQNs since their inception in 2013, highlighting how they have revolutionized autonomous learning across various domains, including robotics, gaming, and industrial automation. It tracks key improvements that have enhanced DQN’s learning stability, efficiency, and applicability, offering a comprehensive overview of the algorithm’s impact on advancing reinforcement learning techniques and enabling more sophisticated autonomous agents [33].

Accurate disease prediction using electronic health records (EHRs) faces challenges such as limited dataset sizes, ethical considerations, and data complexity. This paper suggests a novel deep learning strategy that makes use of deep Q-learning (DQL), a technique that combines neural networks and reinforcement learning, to address these problems. Through efficient management of complicated variable interactions and insufficient patient data, the suggested approach aims to improve prediction accuracy. Experimental results demonstrate that the DQL-based disease prediction model surpasses traditional predictive techniques, achieving an impressive accuracy rate of 98%, which could have significant implications for healthcare diagnostics and personalized medicine [34].

Policy gradient methods are reinforcement learning approaches that use gradient descent to maximize the expected long-term cumulative reward, thereby directly optimizing parameterized policies. They avoid problems including inaccurate value function estimation, difficulties with unclear state information, and the difficulty of managing continuous states and actions, in contrast to classical reinforcement learning [35].

Policy gradient techniques are a key component of deep reinforcement learning, which allows agents to directly learn optimal actions through interaction with their surroundings. This overview presents a thorough examination of on-policy policy gradient algorithms, all derived from the foundational Policy Gradient Theorem but differing in their design and application. The paper provides a rigorous proof of the continuous Policy Gradient Theorem, explores convergence properties, and reviews practical algorithm implementations. It further compares the performance of leading algorithms on continuous control tasks, emphasizing the beneficial role of regularization techniques in improving learning stability and effectiveness [36].

This chapter introduces Artificial Neural Networks (ANNs), which are computational models drawing inspiration from the nervous systems of higher organisms. ANNs comprise artificial neurons that perform mathematical calculations and, through their interconnected structure, can learn to process information to generate desired outputs. They are versatile tools applied to tasks like pattern classification, data mining, function approximation, and information processing across various fields. The chapter focuses on explaining the MultiLayer Perceptron (MLP), the most widely recognized ANN architecture, starting with an introduction to the artificial neuron, which is the fundamental unit of the MLP [37].

Neural networks require activation functions because they determine how artificial neurons interpret inputs to generate outputs and allow the network to recognize intricate patterns. This paper offers a comprehensive and up-to-date overview of activation functions, tracing their evolution from classical options like logistic sigmoid and ReLU to numerous newer variants developed alongside deep learning advancements. The study is a useful tool for researchers and practitioners looking to choose or create efficient activations for different neural network designs and applications since it examines the mathematical characteristics, benefits, and drawbacks of well-known activation functions [38].

The Backpropagation Neural Network (BPNN), a deep learning model first presented in the 1980s and modeled after biological neural networks, is highlighted in this overview. BPNN, which consists of input, hidden, and output layers, uses the backpropagation technique to optimize weights, enabling robust learning and flexibility. Applications including image recognition, speech processing, natural language processing, and financial forecasting make heavy use of it. Its training involves adjusting weights and biases with optimization techniques like gradient descent and metaheuristic algorithms including Grey Wolf, Genetic, Particle Swarm, and Simulated Annealing. With ongoing deep learning advancements, BPNN is expected to remain crucial in key tasks such as image and speech recognition [39].

Optimization algorithms are vital for effectively training neural networks, enhancing convergence speed, and achieving superior accuracy. This thesis explores several core optimizers including classical momentum, Nesterov momentum, AdaGrad, AdaDelta, RMSprop, Adam, AdaMax, and Nadam detailing their mathematical foundations and operational distinctions. Alongside these, it covers essential neural network components such as activation and loss functions to provide a robust theoretical background. The study further focuses on Recurrent Neural Networks (RNNs), which are specialized for capturing temporal dependencies in sequential data, presenting an overview of their architecture. These concepts are practically applied to classify EEG signals corresponding to various meditative states, demonstrating the direct translation of neural network theory into impactful scientific analysis of subjective conscious experiences [40].

This synopsis emphasizes the deep learning technologies' tremendous influence and quick development, as they increasingly mimic human cognitive functions including learning, problem-solving, and decision-making. Convolutional neural networks (CNNs), in particular, are deep learning models that can self-train without requiring a lot of human programming. Applications such as image classification, segmentation, object detection, video processing, natural language processing, and speech recognition make substantial use of CNNs. There are typically four main levels in a CNN architecture.:

Convolutional Layer: Applies kernel filters to the input image to extract fundamental features via convolution operations.

Pooling Layer: Often positioned after convolutional layers, this layer reduces spatial dimensions to consolidate features.

Fully Connected Layer: Also referred to as the output layer, this layer facilitates categorization by connecting each neuron from the previous layer to the one after it.

The non-linear activation layer, which typically uses functions like Sigmoid, Tanh, ReLU, Leaky ReLU, Noisy ReLU, and Parametric Linear Units, defines the output behavior of neurons.

The structure and operation of the visual cortex in the human brain serve as models for CNN architecture, which mirrors how neurons interact and interpret visual data. LeNet, AlexNet, and VGGNet are well-known CNN architectures that have significantly advanced deep learning tasks using images [41].

This study compares popular CNN architectures (LeNet, AlexNet, VGG16, ResNet-50, Inception-V1) for lung cancer detection using the LUNA16 dataset. Various optimizers (RMSProp, Adam, SGD) were tested. AlexNet with SGD achieved the best results with approximately 97.4% accuracy and surpassed the others in key metrics like sensitivity and specificity [42].

The study focuses on Speech Emotion Recognition (SER) utilizing Gated Recurrent Units (GRU), which frequently encounter overfitting issues. To mitigate this, various optimization techniques were evaluated. The most favorable outcomes resulted from combining Dropout (20%), Batch Normalization, and Xavier/Glorot Initialization. This approach enhanced test accuracy from 63.21% to 67.34% by reducing overfitting. The improved model was subsequently integrated into a danger recognition system to augment safety, particularly for users with speech impairments [43].

This work investigates the application of Generative Adversarial Networks (GANs) to create synthetic data from real datasets in educational research. Starting with survey data on university teachers' self-perceptions regarding digital competence and pedagogical knowledge (TPACK model), the study generates 29 synthetic datasets using the COPULA-GAN method. A two-stage cluster analysis compares the synthetic and original data to assess their similarity. Results indicate that the synthetic data closely mirror the original, consistently identifying three teaching profiles based on technical-pedagogical knowledge. The study concludes that synthetic samples offer significant advantages for data quality, anonymization, and expanding sample sizes in research [44].

Hyperparameters are crucial in determining the performance of machine learning models on unseen data. Despite this, a review of 64 political science papers from leading journals (2016–2021) indicates that only about 20% report their hyperparameters and tuning methodologies. This lack of transparency can undermine confidence in model comparisons, as illustrated through an example predicting electoral violence from tweets. The study emphasizes that proper hyperparameter tuning and documentation should be standard practice in assessing model robustness [45].

A peer review of Saeb et al.'s publication, "The necessity to approximate the use-case in clinical machine learning," served as the impetus for this three-part study, which examines the difficulties of cross-validation in clinical machine learning. It discusses the suitability of several cross-validation techniques and how to interpret their results, offering viewpoints from both the original authors and reviewers [46].

Generative models are widely employed to model and produce high-dimensional data resembling real-world data, but tuning their hyperparameters is often time-consuming. This paper proposes an efficient hyperparameter search method that accelerates tuning by adaptively allocating computational resources: it quickly terminates underperforming configurations and concentrates more on promising ones. Framing hyperparameter search as a best-arm identification problem, the method combines hypothesis testing with Successive Halving and utilizes an exponentially weighted Maximum Mean Discrepancy (MMD) metric to compare intermediate model performances. Experiments demonstrate that this approach outperforms traditional Successive Halving by selecting superior hyperparameters across various real-world tasks and budgets [47].

This paper offers a comprehensive exploration of confusion matrices as a primary tool for evaluating machine learning classifiers. It describes the basic elements of confusion matrices and how they support key performance indicators like F1 score, accuracy, precision, recall, sensitivity, specificity, and false positive rate. It also discusses sophisticated evaluation metrics including precision-recall curves, AUC, and ROC curves. Other significant metrics, such as G-mean, Cohen's Kappa, prevalence, null error rate, markedness, average precision, and balanced accuracy, are also covered in the study, emphasizing their use in enhancing and improving classifiers for actual data problems [48].

As Industrial Control Systems (ICS) increasingly integrate with internet technologies, cybersecurity risks have surged, challenging traditional defense mechanisms. A significant oversight in ICS cybersecurity research is the inadequacy of current datasets for training Machine Learning (ML)-driven Intrusion Detection Systems (IDS). To tackle this, the authors introduce ICS-Flow, a comprehensive new dataset containing over 25 million raw network packets, flow records, and process state logs from simulated ICS environments under normal operation and attack conditions. Alongside the dataset, they present ICSFlowGenerator, an open-source tool designed to extract detailed flow parameters from raw packet data. This paper aims to bridge that gap. We show that our dataset, compatible with both supervised and unsupervised ML, effectively trains strong IDS models, as demonstrated by tests using decision tree, random forest, and neural network approaches, specifically designed for ICS protection [49].

The Fourth Industrial Revolution has transformed industrial environments by tightly integrating humans, machines, and internet-connected technologies, drastically increasing the number of connected devices. This article explores the resulting cybersecurity challenges, emphasizing the critical importance of safeguarding cyber-physical systems (CPS), such as Programmable Logic Controllers (PLCs) and SCADA systems, that underpin modern industrial automation. It discusses risk management frameworks, identifies key threat vectors including human, hardware, and software sources and reviews existing security measures within industrial settings. Additionally, the paper proposes enhancements to industrial network architectures and communication protocol security to mitigate risks of sabotage, data breaches, and operational disruptions in increasingly automated and connected industries [50].

Within this guidance, the unique operational, dependability, and safety specifications of Industrial Control System (ICS) environments such as SCADA, DCS, and PLC-based configurations are thoroughly examined. The document proceeds to delineate typical system designs, identify frequently encountered threats and exploitable weaknesses, and then prescribe relevant security countermeasures to lower the associated risks for these essential infrastructures [51].

The cybersecurity landscape for modern power grid SCADA systems is challenged by their ongoing shift from proprietary communication protocols to internet-based ones, such as IEC-60870-5-104. These protocols lack built-in security and are vulnerable to exploitation. To enhance SCADA security, the paper proposes a Security Monitoring Unit (SMU) that uses Deep Packet Inspection (DPI) and white-list rules tailored for IEC-60870-5-104. It also introduces data correlation between sensor values and network data for message validation.

The approach effectively detects both known threats and zero-day attacks in SCADA systems [52].

This study describes a security incident where all death row cell doors opened unexpectedly, prompting an investigation. Many prisons rely on SCADA systems with PLCs to control door mechanisms. Drawing from Stuxnet-related research, it was found that PLCs in correctional facilities have critical vulnerabilities that allow attackers to remotely manipulate door states. By leveraging publicly available exploits and analyzing both electronic and physical security flaws, the study assesses the risks in SCADA-controlled correctional systems and offers recommendations to strengthen security in such high-risk environments [53].

In addition to these systemic issues, Rockwell Automation's Allen-Bradley MicroLogix 1100 and 1400 models were found to have a remote code execution (RCE) vulnerability. The U.S. Department of Homeland Security acknowledged this vulnerability, which was given a critical CVSS v3 base score of 9.8. Presented at the 2015 ICS Cyber Security Conference in Atlanta, the underlying research, which entailed developing custom software, was notable for its unique approach in contrast to previous vulnerabilities in Rockwell equipment. The consequences of this particular vulnerability are significant for the integrity of worldwide ICS security, given the extensive use of these controllers in industrial operations [54].

This report, prepared by the US-CERT Control Systems Security Center (CSSC) at Idaho National Laboratory (INL), aims to:

Document and analyze 120 cybersecurity incidents related to control systems. Support the development of standardized incident reporting and analysis methodologies. Enhance awareness and participation in cybersecurity measures within private and corporate sectors.

The incidents were sourced from various databases and organizations, including:

British Columbia Institute of Technology (BCIT) Industrial Security Incident Database

2003 CSI/FBI Computer Crime and Security Survey

KEMA, Inc. Database

Lawrence Livermore National Laboratory

Energy Incident Database

INL Cyber Incident Database [55].

This paper tackles the cybersecurity challenges confronting Industrial Control Systems (ICS) by introducing an AI-driven Intrusion Detection System (IDS) specifically designed for these environments. The system uses a combination of supervised and unsupervised machine learning approaches to perform real-time anomaly detection and pattern recognition, accurately identifying cyber-attacks. Through experimental evaluations, the IDS has proven highly effective and scalable in detecting threats within real-world ICS settings, providing a strong solution to bolster the security of critical infrastructure [56].

The increasing integration of information technology into ICS, coupled with their connectivity to the internet and cloud computing, has unfortunately opened the door to new vulnerabilities and more sophisticated cyber threats. Traditional security models relying on ICS isolation are no longer sufficient to protect these systems. This highlights the need to enhance ICS cybersecurity using intelligent approaches, particularly machine learning (ML) techniques. ML enables early detection of cybersecurity issues and helps mitigate their impact without causing real damage. Exploring the deployment of ML techniques for intrusion detection within ICS, this paper seeks to inform the optimal selection of ML approaches specifically for anomaly detection in these vital infrastructures [57].

As cyberattacks escalate in both frequency and sophistication, conventional threat detection strategies are increasingly proving insufficient. Artificial Intelligence (AI), with its sophisticated capabilities in data processing and pattern recognition, has emerged as a crucial tool for bolstering cybersecurity defenses. This paper offers an overview of the current landscape of AI in cybersecurity, emphasizing foundational techniques such as machine learning and deep learning as applied to threat detection. It further explores the advantages of integrated learning and multimodal methodologies. The discussion concludes by addressing current obstacles in AI-driven cybersecurity and charting potential future avenues, with the objective of enhancing the precision and operational effectiveness of threat detection mechanisms [58].

In the contemporary digital age, the immense volume of data and the increasing intricacy of cyber threats pose substantial hurdles for cybersecurity. While long-standing signature-based intrusion detection methods have seen widespread adoption, they exhibit inherent limitations when confronting rapidly evolving threats. Artificial Intelligence (AI) methodologies particularly Machine Learning (ML), Deep Learning (DL), and ensemble learning have demonstrated considerable promise in improving the efficacy of attack detection. This comprehensive review synthesizes insights from 72 research papers, classifying AI-based intrusion detection approaches based on their algorithms and performance metrics. It observes that, while AI methods generally boost detection accuracy, the majority of research prioritizes overall detection performance over the detailed classification of specific attack types. The study ultimately seeks to offer a thorough understanding of AI-driven intrusion detection mechanisms and to guide future investigations into the complexities of multi-class attack classification [59].

Cyberattacks on Industrial Control and Automation Systems (ICAS) have surged due to the convergence of Information Technology (IT) and Operational Technology (OT). Historically isolated and running on proprietary protocols, ICAS are now increasingly integrated with smart technologies like IIoT, M2M, Digital Twins, cloud computing, and AI to enhance efficiency. This integration introduces new vulnerabilities exploitable by attackers. The operation of vital infrastructure, such as power plants, nuclear facilities, water utilities, and oil, gas, and manufacturing operations, falls under the purview of ICAS. Consequently, attacks on these systems carry profound risks, including threats to human life, service disruptions, economic damages, and jeopardized national security. While numerous defensive measures exist, absolute security remains elusive; instead, true resilience is built through a layered, defense-in-depth approach. This document reviews the cybersecurity standards and frameworks pertinent to ICAS, explores current threat landscapes and weaknesses, and examines methodologies for securing these indispensable systems [60].

Industrial Control Systems (ICS) comprise the control mechanisms and instrumentation employed for overseeing and managing industrial processes. A fundamental element within ICS is Supervisory Control and Data Acquisition (SCADA), responsible for broad-scale industrial operations spanning diverse geographical areas. Noteworthy cyberattacks, like the Colonial Pipeline ransomware incident, have starkly revealed the profound risks to this infrastructure, resulting in significant disruptions such as fuel shortages. Conventional tools for vulnerability assessment prove insufficient for ICS environments, highlighting a demand for specialized vulnerability datasets. To address this, this paper presents ICS-LTU2022, an exhaustive metadata dataset designed for vulnerability assessment and risk evaluation in ICS. The dataset incorporates detailed features and is analyzed for predominant vulnerabilities based on their severity, frequency, impact, affected components, and common weaknesses. Updated twice a year, ICS-LTU2022 furnishes security researchers with current data on critical ICS vulnerabilities, thereby assisting in the development of enhanced defense strategies [61].

Supervisory Control and Data Acquisition (SCADA) systems constitute the foundational monitoring and control backbone for essential sectors, including energy, telecommunications, transportation, pipelines, chemical processing, and manufacturing. Historically operating in isolation, older SCADA systems faced fewer Internet-borne threats. However, their increasing integration with the Internet and enterprise networks has introduced substantial security complexities. In response to the growing number of security incidents targeting SCADA infrastructure, this survey provides a comprehensive review of SCADA architecture and communication protocols. It spotlights significant security breaches and prevalent threats, and scrutinizes existing security measures. Furthermore, the paper delves into the current state of SCADA security, ongoing research directions, and prospective future enhancements aimed at fortifying protection [62].

Machine learning (ML) techniques are increasingly being utilized to bolster the resilience of Industrial Control Systems (ICS) against cyber-attacks. These methodologies predominantly focus on two critical domains: network-level intrusion detection, which leverages data from network packets, and physical process-level anomaly detection, based on analysis of system behavior data. This survey systematically reviews four primary machine learning paradigms supervised, semi-supervised, unsupervised, and reinforcement learning as they are applied to intrusion and anomaly detection within ICS environments. Selected scholarly works are meticulously analyzed and structured within a seven-dimensional framework to facilitate clear comparison. Intended for researchers, students, and practitioners, this survey identifies prevailing challenges and research gaps, offering actionable recommendations to advance the field [63].

An Intrusion Detection System (IDS) acts as a critical defense layer for Industrial Control Systems (ICS), which operate continuously to manage vital processes. A prominent example is the Supervisory Control and Data Acquisition (SCADA) system, responsible for monitoring and controlling physical processes in critical infrastructure. With cyber threats targeting industrial automation on the rise, machine learning-based detection techniques have become essential for intrusion detection in SCADA environments. Using labeled datasets, these algorithms analyze network traffic between ICS components by inspecting data packets and extracting flow-based features. This approach enables behavior analysis, port-wise profiling, and anomaly detection to classify and predict potential cyber-attacks effectively [64].

The advent of Industry 4.0 has coincided with a sharp increase in cyberattacks targeting Industrial Control Systems (ICS). These systems are increasingly exploited by cybercriminals and state-sponsored actors given their profound impact on industrial operations. While many cyberattack detection systems have been developed, those designed for ICS face distinct challenges not typically found in conventional cybersecurity. This paper aims to achieve three objectives: (1) to explore the present vulnerability landscape of ICS, (2) to review recent progress in machine learning (ML)-based detection methods, with a particular focus on ML classifiers, and (3) to evaluate the strengths and weaknesses of these methods concerning detection accuracy and the range of detectable attack types. Drawing from these insights, the paper identifies significant open challenges and highlights promising research avenues for the community [65].

This paper outlines the development and deployment of a SCADA system augmented for automated fault detection. This system offers three primary advantages: it facilitates preventive and predictive maintenance through prognostic capabilities, elevates product quality, and significantly reduces periods of machine downtime. By integrating Industrial Internet of Things (IIoT) technologies and diverse machine learning (ML) techniques, the system analyzes a variety of data sources. Notably, it replaces certain digital sensors with analog ones to improve the detection of faults with ambiguous origins. Furthermore, a novel anomaly detection algorithm is incorporated to forecast failures and identify malfunctions that do not trigger standard alarms. The demonstrated improvement in machine availability following this system's implementation attests to its effectiveness in achieving these stated objectives [66].

The escalating number of successful cyberattacks on Industrial Control Systems (ICS) has created an urgent demand for precise and timely anomaly detection to safeguard critical processes. Data-centric anomaly detection methods that leverage machine learning have garnered considerable attention due to their inherent ability to automatically learn the dynamics and control strategies of ICS processes. These approaches streamline the development of detectors, making them faster and easier to create compared to traditional, physics-based design methods. However, considerable challenges persist in the creation and deployment of machine learning-based detectors, particularly within large-scale industrial plants. This work identifies and discusses these ongoing challenges and shares valuable lessons learned from the practical implementation of such detectors in an operational plant environment [67].

This survey addresses the important problem of anomaly detection by providing a structured and comprehensive overview of deep learning–based methods. It categorizes state-of-the-art research techniques according to their underlying assumptions and approaches. For each category, the survey explains the basic anomaly detection method, its variants, and key assumptions used to distinguish normal from anomalous behavior. It also discusses the advantages, limitations, and computational complexity of these techniques in real-world applications. Additionally, the survey reviews how these methods have been adopted across various domains and evaluates their effectiveness. Finally, it highlights open research issues and challenges in applying deep learning to anomaly detection [68].

Modern industrial control systems (ICS) are increasingly connected to corporate Internet networks to leverage online resources. However, this heightened connectivity exposes ICS to a wider range of cyberattacks, making their security a critical concern. Intrusion detection technology serves as a vital security measure by effectively identifying potential attacks on ICS. This survey first explores the unique characteristics and evolving security requirements of ICS. A novel taxonomy for Intrusion Detection Systems (IDS), specifically tailored for ICS, is then introduced, grouping them into protocol analysis-based, traffic mining-based, and control process analysis-based methods. The survey proceeds to weigh the benefits and drawbacks of each approach, concluding with a discussion on future research trajectories for advancing intrusion detection in industrial control system [69].

The escalating importance of cyber-physical systems (CPS), and more specifically Industrial Control Systems (ICS), in critical sectors like electricity, oil and gas, water, chemical processing, and healthcare, has significantly propelled cybersecurity to the forefront of attention recently. This document initially clarifies the core concepts of CPS, underlining the integral function of wireless sensor networks (WSNs). It then proceeds to review the existing landscape of ICS security in the United States, detailing various initiatives from both government and industry, including management strategies, technological developments, compliance standards, regulatory measures, and research conducted by national laboratories. Next, it highlights European ICS security efforts, focusing on a key ENISA report. Finally, the paper contrasts these developments with the challenging ICS security landscape in China and outlines the country's ongoing security management efforts [70].

In automated manufacturing, networks of sensors and actuators communicate via industrial fieldbuses like PROFIBUS, connecting automation units and supervisory systems. With the rise of Industry 4.0, upgrading legacy systems is a key challenge. To modernize a legacy Flexible Manufacturing System (FMS), this paper proposes a joint hardware and software approach. This solution enhances connectivity by enabling Ethernet communication among its sensors, actuators, and supervisory systems. Test results not only demonstrate the upgraded FMS's efficient operation but also affirm the viability of this strategy, signifying progress toward Industry 4.0 integration [71].

Anomaly detection is vital for securing cyber-physical systems (CPS). However, the growing complexity of CPSs and increasingly sophisticated attacks challenge traditional detection methods, which often require domain-specific knowledge and struggle with large data volumes. To address this, deep learning-based anomaly detection (DLAD) methods have emerged. This paper reviews the latest DLAD techniques for CPS, proposing a taxonomy based on anomaly types, strategies, implementation approaches, and evaluation metrics to clarify their key features. Using this framework, we highlight unique designs across different CPS domains, discuss existing limitations and open challenges, and provide experimental insights into typical neural models, workflows, and performance. Finally, we outline the shortcomings of current DL methods and suggest directions to enhance DLAD effectiveness, encouraging further research in this area [72].

Catastrophic incidents such as pipeline explosions, halts in production, widespread traffic disarray, railway collisions, nuclear facility shutdowns, extensive power outages, and critical interruptions to ICU oxygen all these can stem from malfunctions in SCADA or Industrial Control Systems (ICS). SCADA systems are indispensable for automating the supervision and control of Critical Infrastructures (CI). While earlier SCADA deployments maintained a lower vulnerability profile due to their operational isolation, modern SCADA systems have developed into complex, internet-linked, and geographically distributed networks, consequently confronting a heightened level of cyber threats. This review paper examines SCADA system architectures and communication protocols, analyzes cyber-attacks targeting these systems, and reviews current intrusion detection techniques and SCADA testbeds. It also explores cloud- and IoT-based SCADA architectures. Finally, the paper identifies critical xresearch challenges to improve SCADA security and close existing vulnerabilities [73].

1.2 PROBLEM STATEMENT

In our work we have several datasets for industrial control system cyberattacks. For detection of the threats firstly we apply multiclassification methods: AdaBoost, Random Forest, K-Nearest Neigbous, Decision Tree, Gradient Boosting. Then apply PCA method for reducing of our data. We apply AI for determination of measures in confusion matrix as we work with large datasets. In our work as example we consider a small dataset, apply the multiclassification using K-Nearest Neighbors method, then we apply PCA method and construct a confusion matrix. Also we show the codes for large datasets using AI methods.

CHAPTER 2. APPLICATIONS OF ARTIFICIAL INTELLIGENCE IN INDUSTRIAL CONTROL SYSTEMS SECURITY & THREAT DETECTION

2.1 Industrial Control Systems (ICS)

Industrial Control Systems (ICS) broadly categorize the control systems and instrumentation crucial for managing and automating industrial operations. These systems are foundational to modern society, overseeing essential services like electrical power generation and delivery, petroleum and gas refining and transport, water distribution and treatment, chemical processing, pharmaceutical production, and critical transportation networks such as railways and air traffic control. Because these applications are inherently mission-critical, they demand exceptionally high availability. Any disruption to an ICS can lead to severe consequences, ranging from major production losses and equipment damage to environmental harm and even threats to public safety [1, 3].

ICS environments feature a diverse array of equipment, integrated systems, networks, and mechanisms. Key configurations within the ICS landscape include Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and Programmable Logic Controllers (PLC). The National Institute of Standards and Technology (NIST) offers comprehensive guidance on securing these complex systems, recognizing their unique operational performance needs and the necessity for tailored security approaches [1].

Supervisory Control and Data Acquisition (SCADA) systems are designed to monitor and control industrial processes across broad geographical areas. They provide operators with a graphical user interface (GUI) that displays the status of the system, allows for the receipt of alarms indicating abnormal operations, and enables real-time adjustments to manage the process. SCADA systems centralize data acquisition and supervisory control functions for geographically dispersed assets [4].

Distributed Control Systems (DCS) are characterized by an architecture where controllers, sensors, and actuators are dispersed across multiple physical sites. Within a DCS, numerous controllers work collaboratively to manage the various sub-components of the overarching system. These systems are specifically engineered for highly automated and intricate loop control, often leveraging microprocessor technology to execute complex computational algorithms and logical expressions [1].

Specialized industrial computer control systems are called *Programmable Logic Controllers or PLCs*. In order to efficiently manage output devices, they must constantly monitor input devices and make decisions based on a customized software. PLCs hold instructions for a wide variety of tasks, including as logic operations, sequence control, timing, counting, and arithmetic computations, in programmable memory. They do this by using their digital or analog input and output interfaces to monitor production processes and mechanical equipment [1].

Core Components of ICS Architecture

An industrial control system's architecture is essentially made up of interconnected hardware and software components intended to make automation, control, and monitoring easier. Built on top of multilayer network architectures that make use of a variety of communication protocols, a typical ICS includes several control loops, remote diagnostics capabilities, maintenance tools, and human interfaces.

At the lowest level, directly interacting with the physical process, are **sensors**. These devices detect changes in the environment, such as variations in pressure or temperature, and transmit this information as controlled variables to a controller. Their role is critical for gathering real-time operational data and they constitute Level 0 in the Purdue Model. Complementing sensors are **actuators**, often referred to as movers. These are mechanical elements designed to manipulate or regulate a process, such as control valves, circuit breakers, switches, and motors. Actuators receive output variables from controllers, translating digital commands into physical actions to adjust the industrial process. Like sensors, they also reside at Purdue Model Level 0.

Controllers, which include devices like PLCs and Remote Terminal Units (RTUs), receive input from sensors, execute predefined control algorithms, and generate the necessary output variables to command actuators. These components are vital for interpreting raw data from Level 0 and executing commands, ensuring the industrial process operates safely and efficiently; this positions them at Level 1 of the Purdue Model. Within a Distributed Control System (DCS), a specialized "process control unit" acts as the central processing element, performing all necessary computation algorithms and logical expressions to manage control loops [1].

Human-Machine Interfaces (HMIs) provide the crucial connection between human operators and automated systems. These user interfaces display real-time process status, allowing operators to monitor and configure controller parameters, observe ongoing processes, and access process variables, control parameters, and alarms. HMIs are located at Level 2 (Local Control and Supervision) in the Purdue Model, facilitating human interaction with automated systems [1, 4, 5].

Data Historians are centralized databases designed to store all process information generated within an ICS environment. The substantial volume of data recorded by these systems is often transferred to corporate Information Systems (IS). This transfer facilitates comprehensive process data analysis, supports control optimization efforts, and aids in strategic planning. Typically, Data Historians are situated at Level 3 (Site-wide Control and Management) within the Purdue Model [1, 4].

Communication Systems and Protocols are indispensable for facilitating the transfer of data among the various components and across different network segments within an ICS. Common network protocols include Ethernet, Profibus, and DeviceNet. Historically, fieldbus protocols were standardized in the mid-1980s, a period predating widespread internet connectivity, meaning they were designed with little to no inherent security beyond physical access control. In contrast, modern ICS increasingly rely on Internet Protocol (IP) for broader connectivity [1][6].

Within a Distributed Control System (DCS), an Engineering Workstation acts as the overarching supervisory controller. It provides specialized configuration tools, enabling users to perform critical tasks like creating new control loops, setting up input/output (I/O) points, and configuring distributed devices across the system. These workstations are standard features in many Industrial Control System (ICS) architectures. Likewise, Operator Stations serve as dedicated hubs where users can monitor the ongoing industrial process, accessing real-time process variables, control parameters, and alarms vital for understanding the current operational status [1, 5].

The inherent complexity of ICS architecture, characterized by its diverse components and layered structure, significantly contributes to the cybersecurity challenges these systems face. ICS involves a wide array of specialized hardware, including sensors, actuators, PLCs, and RTUs, alongside critical software such as HMIs, SCADA systems, and complex control algorithms. These components operate across multiple hierarchical levels, from the physical processes at the plant floor to enterprise-level business systems, necessitating intricate communication protocols to ensure seamless operation. A critical factor is that many ICS components, particularly legacy systems, were not originally designed with modern cybersecurity in mind. They were developed under the assumption of operating within isolated, trusted networks, where external threats were not a primary concern. This fundamental design philosophy, coupled with the sheer number and variety of interconnected components, creates a vast and heterogeneous attack surface. Each component, especially those with web-accessible interfaces or critical control functions like HMIs, SCADA systems, and PLCs, can become a potential point of vulnerability. This intricate landscape mandates a multi-faceted security approach that extends beyond traditional IT security paradigms. It requires specialized knowledge of Operational Technology (OT) environments and their unique constraints, such as stringent real-time operational demands, the prevalence of proprietary systems, and challenging environmental factors. The integration of diverse technologies and the imperative for interoperability further complicate the security landscape, making it difficult to apply standardized security practices uniformly across the entire system [1][6].

The Purdue Enterprise Reference Architecture (PERA)

The Purdue Enterprise Reference Architecture (PERA), widely recognized as the Purdue Model, emerged in the 1990s as a cornerstone reference model for enterprise architecture. Its development was a collaborative effort involving Theodore J. Williams and members of the Industry-Purdue University Consortium for Computer Integrated Manufacturing. The core objective behind its creation was to establish best practices for structuring and managing Industrial Control Systems (ICS) and their interconnections with business networks. This involved defining systematic approaches for isolating Operational Technology (OT) systems from corporate Information Technology (IT) systems. A central tenet of the

Purdue Model is the distinct separation of OT from IT environments. This segregation aims to facilitate effective communication management by isolating disparate functions across discrete layers and implementing robust network segmentation. Initially, this separation was often conceptualized as "air-gapping," implying a physical disconnection of ICS and OT systems from the internet to ensure their security.

The Purdue Model systematically organizes an industrial control system into multiple layers, each assigned specific roles and delineated communication boundaries:

Level 0: Physical process: This foundational layer encompasses the physical devices that directly interact with industrial processes, such as sensors and actuators. Its primary role is to collect and transmit real-time data on process variables like pressure and temperature. While essential for operational data, this layer is inherently susceptible to physical attacks or system malfunctions.

Level 1: Intelligent devices: This layer consists of intelligent devices that monitor and control industrial processes, such as Remote Terminal Units (RTUs) and Programmable Logic Controllers (PLCs). To guarantee safe and effective functioning, they analyze Level 0 data and carry out directives. Although this level is essential for control, it is susceptible to cyberattacks, frequently as a result of outdated hardware and low processing power.

Level 2: Local oversight and management Human-Machine Interfaces (HMIs) and SCADA systems are examples of supervisory control systems found in this tier. Its functions include process supervision, alarm management, and real-time adjustment by human operators. This level, which is usually in charge of particular areas of a facility, acts as the interface between automated systems and human operators.

Level 3: Management and control of the entire site: This layer oversees more general operational tasks like production scheduling and general operational analysis and is made up of systems like data historians and alarm servers. It is positioned strategically as the first line of defense between the enterprise IT network and the process control environment.

Level 3.5: Industrial DMZ (iDMZ) and Demilitarized Zone (DMZ): Despite not being included in the original Purdue Model, the DMZ is now a crucial component of contemporary ICS security. Between the IT and OT environments, it functions as a buffer zone. By filtering and controlling traffic flow, it protects vital OT systems from unwanted access while allowing for essential data interchange. Between the IT and OT network segments, this area acts as a vital enforcement boundary.

Level 4: Business logistics systems: These systems are located at the enterprise level and are connected to the larger corporate network. Examples of these systems are CRM and ERP. In order to support higher-level decision-making and business planning, it integrates industrial data and makes sure that industrial operations are in line with overall business strategies.

Level 5: Business network All typical corporate IT operations, including file storage and email, are managed by this topmost layer. It supports basic corporate operations and may request data from lower levels for business analytics, but usually not having direct interface with industrial control system components.

The Purdue Model holds significant importance for network segmentation and security within industrial environments. It offers a structured methodology for balancing operational efficiency with cybersecurity by clearly defining and segmenting ICS and IT systems. This segmentation is a fundamental element of IT/OT security, as it helps enforce logical separation, which is crucial in preventing the lateral movement of attackers if one system is compromised. Consequently, it effectively minimizes the "blast radius" of a cyberattack. The model has evolved into an influential tool for determining where to implement security controls and how to effectively monitor for threats. Its foundational design has been adopted by various industrial control system frameworks, including NIST 800-82 and API 1164 [7].

Evolution and Modern Trends

The architecture of ICS is continuously evolving due to advancements in technology and the changing demands of industrial operations:

Increased Connectivity and IT/OT Convergence: Greater integration between ICS (Operational Technology - OT) and enterprise IT systems to enhance data sharing and business intelligence [8]

Industrial Internet of Things (IIoT) Adoption: Integration of smart sensors, cloud computing platforms, and advanced analytics to enable enhanced monitoring, predictive maintenance, and operational efficiency.

Growing Emphasis on Cybersecurity: Increasing awareness of cyber threats targeting ICS has led to the implementation of robust security measures across all architectural levels.

Wireless Communication Technologies: The increasing use of wireless communication protocols within the plant floor offers greater flexibility and reduces wiring complexities .

Cloud-Based ICS Solutions: The emergence of cloud platforms for data storage, analytics, and, in some cases, control functionalities within ICS architectures.

SCADA Systems: Architecture, Functions and Use

Power grids, water utilities, transportation networks, manufacturing facilities, and oil and gas operations are just a few of the industries that depend heavily on Supervisory Control and Data Acquisition (SCADA) systems for monitoring and controlling industrial processes [9, 11]. These systems empower organizations to remotely or locally control industrial processes, continuously monitor, collect, and process real-time data, and leverage this information for strategic decision-making to ensure operational efficiency and minimize downtime [11].

Architecture of SCADA Systems

A typical SCADA system comprises several key components working together to provide comprehensive monitoring and control capabilities [9, 15]:

Remote Terminal Units (RTUs) and Programmable Logic Controllers (PLCs) function as essential field devices, linking directly to sensors and actuators within an industrial process. They gather crucial data from sensors like temperature, pressure, and flow rate and manage actuators, such as valves, pumps, and motors, in response to commands from a master station [9]. Increasingly, modern systems are integrating IoT devices to further enhance data acquisition and communication capabilities [15].

The communication network is vital for enabling data exchange between the RTUs/PLCs and the master station. While various communication protocols are employed, contemporary SCADA systems are progressively moving towards open, internet-connected architectures [9].

Master Terminal Unit (MTU) or SCADA Server: This is the central component that collects and processes data received from the RTUs/PLCs. It provides a human-machine interface (HMI) for operators to visualize the process, monitor alarms, and issue control commands [9, 10]. The MTU often includes a historical database (Historian) to store and retrieve process data for analysis [9].

The Human Machine Interface (HMI) gives operators a graphical way to communicate with the SCADA system. It lets operators give control commands and shows data, trends, and alarms in real time. The SCADA system gathers historical process data, which is stored in a centralized database called Data Historian. In order to increase operational efficiency, this data is essential for analysis, reporting, and trend identification [9, 10]. The architecture of SCADA systems has evolved from standalone systems to more complex, distributed, and networked systems [9]. Modern trends focus on integrating web servers and OPC (OLE for Process Control) to enhance interoperability and remote access [13]. Architectures are also being designed for improved interoperability, reduced complexity, and enhanced maintainability [16]. For microgrids, SCADA systems act as a central hub for intelligent monitoring, enabling communication and control between local controllers and upper web monitoring systems [14].

Functions of SCADA Systems

SCADA systems perform a wide range of functions essential for the operation and management of industrial processes [10, 15]:

Data Acquisition: Gathering real-time data from remote sites through RTUs/PLCs and sensors.

Supervisory Control: Enabling operators to monitor and control remote equipment and processes through the HMI [9, 11]. This includes sending commands to actuators to adjust process parameters.

Data Presentation: Displaying collected data and process status to operators in a user-friendly graphical format via the HMI. This allows for real-time visualization and understanding of the process.

Alarm Management: Detecting and notifying operators of abnormal conditions or equipment malfunctions through visual and audible alarms [12].

Data Logging and Historian: Storing historical process data for analysis, reporting, and trend identification [9, 10]. This historical data is vital for performance optimization and troubleshooting.

Reporting: Generating reports on process performance, alarms, and historical trends.

Communication: Managing the communication infrastructure and protocols for data exchange between the master station and remote sites [12].

Control Algorithms: Implementing automatic control algorithms to optimize processes and respond to changing conditions [9].

Use of SCADA Systems

In many businesses where it is essential to monitor and manage regionally distributed assets, SCADA systems are invaluable [9, 11]:

Power Generation and Distribution: Keeping an eye on and managing transmission lines, substations, and power plants to guarantee grid stability and effective power distribution [16].

Water and Wastewater Management: Monitoring and managing wastewater collection systems, distribution networks, and water treatment facilities [9].

Oil and Gas: Monitoring and controlling pipelines, refineries, and offshore platforms [9].

Transportation: Managing traffic control systems, railways, and airport operations [9].

Manufacturing: Monitoring and controlling industrial automation processes [9].

Telecommunications: Supervising network infrastructure and equipment [9].

Environmental Monitoring: Collecting data from remote environmental sensors [9].

Cybersecurity Challenges and Architectural Considerations in ICS

Operational Technology (OT) environments, which encompass Industrial Control Systems, present a distinct set of security challenges that differentiate them from traditional Information Technology (IT) landscapes. These challenges stem from the unique operational characteristics and historical development of ICS.

One of the foremost challenges is the real-time operational demands inherent in ICS environments. These systems control physical processes where any delay or disruption can lead to catastrophic consequences, including equipment damage, production loss, or even public safety hazards. This critical requirement makes the introduction of conventional security solutions, such as regular updates or comprehensive scans, significantly more complex than in IT systems, as they could potentially interfere with continuous operation.

A critical issue is the widespread reliance on outdated ICS infrastructure. Many existing systems were built without modern cybersecurity features and, despite their original isolation, now present major vulnerabilities as they connect to corporate networks for remote access. These legacy systems often lack support for contemporary security protocols or patches, making it extremely difficult, expensive, or impossible to secure them adequately without a complete replacement [6].

The widespread use of proprietary systems and protocols further complicates ICS security. Industrial environments often utilize a diverse array of vendor-specific hardware, software, and communication protocols, such as Modbus or DNP3. Many of these protocols were designed primarily for functionality and efficiency, rather than with security features like encryption or authentication. This heterogeneity and lack of standardized security mechanisms make it challenging to apply uniform security practices and integrate with existing security tools [6].

Environmental constraints also play a role in ICS security. Industrial settings are frequently harsh, characterized by electrical noise, dirt, or extreme temperatures. These physical conditions directly impact the feasibility and effectiveness of deploying security solutions. Hardware-based network security tools, typically designed for cleaner, more controlled IT environments, are difficult to implement in such demanding ICS settings. The continuous operational imperative in these environments severely limits opportunities for downtime, turning routine security maintenance and updates into a major logistical hurdle.

The increasing connectivity, driven by Industry 4.0 and IIoT, has led to a substantial expanded attack surface for ICS. Connecting more devices to networks inherently introduces new vulnerabilities and security threats. The widespread adoption of connected devices and sensors in Industry 4.0 creates a greater number of entry points for cyber adversaries seeking unauthorized access, necessitating robust security measures to protect data transmitted across these numerous access points. This expanded attack surface also includes vulnerabilities introduced by human factors, such as employees connecting infected portable devices like USB drives or laptops to industrial systems [3][18].

Identified Vulnerabilities and Architectural Weaknesses

Empirical research has shed light on the most vulnerable components within Industrial Control Systems and the underlying architectural weaknesses that contribute to these susceptibilities. Studies consistently indicate that Human-Machine Interfaces (HMIs), SCADA configurations, and Programmable Logic Controllers (PLCs) are among the most frequently affected components in ICS. For instance, HMIs were cited in 257 vulnerability reports, SCADA software in 212, and PLCs in 114. These components are critical for operator interaction, process control, and data acquisition, making them primary targets for cyberattacks due to their direct interface with both human operators and physical processes.

A significant finding from vulnerability analyses is that a substantial majority 62.86% of vulnerability disclosures in ICS were attributed to an architectural root cause. This critical observation indicates that many ICS products were originally designed without inherent security considerations, operating under the flawed assumption that they would always function within a protected, isolated network environment. The most common architectural weaknesses identified include:

CWE-20: Improper Input Validation (accounting for 26.29% of reports detailing architectural flaws): This represents the most widespread architectural vulnerability, arising when an ICS component either fails to validate incoming inputs correctly or performs the validation improperly. Such deficiencies can be leveraged to initiate denial-of-service (DoS) attacks, thereby disrupting vital industrial operations.

CWE-287: Improper Authentication (identified in 70 reports): This frequent design flaw occurs when an ICS component inadequately or incorrectly verifies the identity claims of an entity attempting to interact with it. This vulnerability often stems from either the absence of appropriate authentication mechanisms or an excessive reliance on the assumption of a protected network, consequently permitting unauthorized access.

CWE-284: Improper Access Control (recorded in 63 reports): This weakness manifests when an ICS component fails to, or incorrectly, restrict unauthorized individuals or systems from accessing resources or equipment. Exploiting this vulnerability can lead to privilege escalation, the unauthorized disclosure of sensitive data, or the execution of malicious code within the control system.

Other significant architectural weaknesses frequently encountered include Cross-site Scripting (CWE-79), Cross-Site Request Forgery (CWE-352), the Use of Hard-coded Credentials (CWE-798), the Use of Hard-coded Passwords (CWE-259), Insufficiently Protected Credentials (CWE-522), and Code Injection (CWE-94). Collectively, these weaknesses underscore a pervasive issue where ICS components are often not built with security as a fundamental design principle. This oversight results in critical vulnerabilities, particularly when these systems are exposed to modern, interconnected threat environments.

The persistent architectural security weaknesses in ICS, coupled with the rapid IT/OT convergence, create a critical and escalating risk for national infrastructure, demanding a fundamental re-evaluation of cybersecurity research and industry collaboration. A substantial majority of ICS vulnerabilities, specifically 62.86%, originate from architectural root causes, largely because these systems were designed without security as a primary consideration, operating under the assumption of isolated networks. This foundational flaw means that common vulnerabilities such as improper input validation, authentication failures, and access control issues are pervasive across ICS components[4].

Simultaneously, ICS environments are undergoing a rapid transformation from isolated systems to highly interconnected ones, integrating with IT networks, cloud platforms, and Industrial Internet of Things (IIoT) technologies. This increasing exposure significantly expands the attack surface and directly introduces vulnerabilities typically found in IT systems into the critical OT domain. Despite the growing sophistication of cyber threats and a notable increase in attacks on critical infrastructure—for instance, energy utilities experienced a 70% rise in cyber intrusions between January and August 2024 compared to the previous year—there remain significant gaps in security research. These gaps include a lack of empirical evidence proving the effectiveness of many cyberdefensive measures, an absence of publicly available data repositories necessary for reproducibility, and limited access to real-world testbeds for academic researchers[4, 6] [18].

The combination of fundamental design flaws in legacy ICS architectures and their increasing exposure through IT/OT convergence creates a fertile ground for sophisticated cyberattacks. The inherent trade-off between usability and security in system design, where access itself can often become a source of vulnerability, further exacerbates this challenge. The current state of security research, hampered by proprietary systems and a lack of data sharing, struggles to keep pace with this rapidly evolving threat landscape. This situation leads to a critical and escalating risk for national infrastructure. It implies that a reactive, patch-based approach is insufficient to address these systemic issues. A proactive, collaborative paradigm shift is required, emphasizing "security-by-design" from the outset of system development. This includes fostering greater industry-academia collaboration to facilitate data sharing and access to testbeds, and investing in advanced, empirically validated security solutions such as AI/ML for anomaly detection and robust architectural segmentation. The evidence clearly indicates a need for a major reevaluation of the founding principles in cybersecurity to adequately protect these vital systems [3][19].

Secure Architectural Design Principles and Countermeasures

To mitigate the escalating cybersecurity risks in ICS environments, the implementation of robust architectural design principles and countermeasures is paramount. A primary strategy involves advanced network segmentation, including microsegmentation and the adoption of zero trust models.Most organizations implement robust perimeter defenses around their Operational Technology (OT) and Industrial Control System (ICS) networks. This is typically achieved by segmenting these networks from their IT counterparts, employing tools such as next-generation firewalls and/or data diodes. This strategy significantly reduces the likelihood that intrusions originating from a compromised IT network could spread into the crucial OT domain. Network segmentation is paramount for enforcing logical separation and considerably limiting the "blast radius" of a cyberattack, thereby containing potential damage. Contemporary security practices advocate for defining and segmenting devices and assets based on criteria like data flows, physical location, critical functionality, or assigned trust levels [3, 7].

Microsegmentation represents an advanced network security technique that further partitions an environment, offering granular visibility into all assets within the same broadcast domain. This enables highly precise control over both "north-south" (client-server) and "east-west" (server-to-server) traffic flows. Complementing this, Zero Trust security models operate on the foundational principle of "never trust, always verify." These models assume that a network breach is inevitable and, consequently, mandate the verification of all identities and devices. They also enforce the principle of least privilege and integrate continuous monitoring and response capabilities across the entire network.

A key recommendation for robust ICS security architecture is the creation of an Industrial Demilitarized Zone (iDMZ). This iDMZ serves as a major enforcement boundary between IT and OT network segments. Firewalls are typically deployed within this zone for boundary protection and to meticulously control information flows and connections between network segments. It is critical to implement stringent firewall rules for both inbound and outbound connections to prevent unauthorized access and data exfiltration. Additionally, unidirectional gateways, also known as data diodes, offer an enhanced layer of protection by physically enforcing traffic flow in only one direction. This safeguards critical OT systems from compromises originating at higher, less secure IT levels.

Crucially, there is an increasing emphasis on security-by-design. Given the criticality of ICS, these systems must be designed from the ground up with security embedded into their core architecture, rather than attempting to retrofit security measures onto legacy systems later. This proactive approach involves integrating security considerations from the initial design phases, encompassing aspects such as end-to-end security, robust access control mechanisms, strong authentication protocols, effective key management, and secure remote monitoring capabilities for IIoT devices [3].

Advanced Cybersecurity Measures and Research Gaps

The dynamic evolution of Industry 4.0 makes it increasingly challenging to predict and identify security gaps through traditional methods. This necessitates the adoption of advanced cybersecurity measures, particularly the application of Artificial Intelligence (AI) and Machine Learning (ML) for anomaly detection. AI and ML techniques are increasingly leveraged to automate cybersecurity processes, learning patterns in ICS signals and data to inform better security decisions and diagnose potential access points for cyberattacks. The efficiency of deep learning models in identifying anomalous behaviors in ICS environments has been shown in numerous research. By continuously monitoring sensors to spot sudden changes that can indicate an attack attempt or a system malfunction, anomaly detection systems are essential to Industrial Control Systems (ICS). Given the inherent challenge of acquiring labeled attack data in authentic industrial environments, unsupervised learning approaches are frequently favored within this domain [3, 7].

To thoroughly evaluate and reduce risks within intricate and legacy Operational Technology (OT) environments, new threat modeling frameworks are under development. Among these are hybrid frameworks that blend approaches like risk-centric (like PASTA), attacker-centric (like Attack Trees), and system-centric (like STRIDE). The risk-centric framework PASTA (Process for Attack Simulation and Threat Analysis) is especially well-suited for intricate, older ICS. It encourages close cooperation between security professionals and operational stakeholders and ranks threats according to their possible business impact. Spoofing, tampering, repudiation, information disclosure, denial-of-service, and elevation of privilege are among the possible system threats that are identified and categorized using STRIDE. Attack trees give a visual depiction of how attackers might take advantage of vulnerabilities and provide a methodical way to describe threats and responses. These frameworks frequently integrate the Common Vulnerability Scoring System (CVSS) for quantitative security assessment and incorporate the MITRE ATT&CK framework for enriched threat intelligence.

Despite these advancements, significant research gaps persist in the field of ICS cybersecurity. There is a notable lack of empirical research proving the effectiveness of many proposed cyberdefensive measures, with "little to no" studies demonstrating their efficacy. Furthermore, often no data repositories are provided, hindering the reproducibility and verification of research findings. A salient risk arises from unexplored vulnerabilities in many merchant-market PLCs, as attacks and defenses have typically been evaluated on only a small subset of the PLCs available globally. Sharing ICS security research is inherently challenging due to the traditionally closed nature of these systems, the prevalence of proprietary hardware and software components, and the significant variability across ICS platforms, which makes generalizable research difficult. Academic researchers frequently struggle to gain access to and modify emerging SoftPLC-based ICS testbeds, which are typically managed by industry experts. Additionally, there is a scarcity of available real Cyber-Physical System (CPS)-generated datasets, largely confining experiments to testbed or simulation environments [3][18, 19].

Case Studies of ICS Security Incidents

Analyzing past security incidents provides valuable insights into the vulnerabilities of Industrial Control Systems (ICS) and the potential consequences of successful cyberattacks. Academic research and industry reports highlight several notable cases that underscore the importance of robust security measures.

Stuxnet (2010): Widely regarded as the first publicly known digital weapon, Stuxnet targeted Iran's nuclear program. This sophisticated malware specifically aimed at Siemens Programmable Logic Controllers (PLCs) used in uranium enrichment centrifuges[20, 21].

Impact: Stuxnet manipulated the rotational speed of the centrifuges, causing physical damage and significantly hindering Iran's nuclear capabilities. It also masked its actions from operators by displaying normal operational values, making detection difficult.

Significance: Stuxnet demonstrated that cyberattacks could have significant physical consequences in the real world and highlighted the vulnerability of industrial control systems to nation-state-level threats.

Maroochy Water Services (2000): This early incident involved a disgruntled former employee who used a laptop and radio transmitter to remotely access the wastewater treatment plant's SCADA system in Queensland, Australia.

Impact: The attacker repeatedly released millions of liters of raw sewage into local parks and rivers over several months.

Significance: This case illustrated how even relatively unsophisticated attacks could cause significant environmental damage and highlighted the importance of access control and insider threat management in ICS environments.

Colonial Pipeline Ransomware Attack (2021): A ransomware attack in 2021 targeted the IT systems of Colonial Pipeline, a primary fuel pipeline operator in the United States. [23]

Impact: Although the initial compromise affected the IT infrastructure, the company preemptively shut down its operational technology (OT) systems. This measure aimed to prevent the ransomware from propagating into the control network, resulting in widespread fuel shortages across the Eastern U.S.

Significance: This incident clearly demonstrated the inherent interconnectedness between IT and OT environments, illustrating how attacks on one can trigger cascading operational failures in critical infrastructure. It also emphasized the increasing threat posed by ransomware to Industrial Control Systems (ICS). [24]

Oldsmar Water Treatment Facility (2021): In 2021, an operator at a water treatment plant in Oldsmar, Florida, observed unauthorized remote access to the plant's Human-Machine Interface (HMI).

Impact: The perpetrator briefly elevated the concentration of sodium hydroxide (lye) to a hazardous level. This action could have resulted in toxic contamination had the operator not immediately intervened and corrected the setting.

Significance: This near-miss incident highlighted vulnerabilities in remote access security and the potential for attackers to directly manipulate critical process parameters with potentially harmful consequences.

Ukrainian Power Grid Attacks (2015 and 2016): These coordinated cyberattacks targeted Ukraine's power grid, resulting in temporary power outages for hundreds of thousands of people.

Impact: The attackers used sophisticated methods, including spear-phishing to gain initial access, deploying BlackEnergy and Industroyer malware to manipulate control systems, and even disrupting recovery efforts by wiping hard drives and targeting communication infrastructure.

Significance: These attacks demonstrated the potential for well-resourced adversaries to cause significant disruption to critical national infrastructure and highlighted the evolving sophistication of threats targeting ICS.

These case studies, among others, emphasize several recurring themes in ICS security incidents:

Interconnectedness of IT and OT: Attacks often leverage vulnerabilities in IT systems to pivot to OT networks.

Legacy Systems: Many ICS rely on outdated hardware and software with known security weaknesses.

Weak Authentication and Access Control: Insufficient password policies, lack of multi-factor authentication, and overly permissive access can be exploited.

Remote Access Vulnerabilities: Improperly secured remote access points can provide entry for attackers.

Insider Threats: Both intentional and unintentional actions by insiders can lead to security breaches.

Evolving Threat Landscape: Attackers are becoming increasingly sophisticated, developing custom malware and tactics specifically for ICS environments.

2.2 Artificial Intelligence Fundamentals

Artificial Intelligence (AI) stands as a revolutionary domain within computer science, focused on creating systems that can execute tasks typically demanding human cognitive abilities. This wide array of tasks includes learning, logical reasoning, intricate problem-solving, perception, and comprehending language. AI systems are designed to process external information, derive knowledge from it, and then apply that knowledge to accomplish defined objectives through adaptable behavior.

Definition and Scope

Defining Artificial Intelligence (AI) can be complex, as its interpretation often shifts between technical and policy discussions. However, the core concept generally remains consistent: AI involves machines simulating human cognitive abilities. It's a broad field, drawing together diverse disciplines including computer science, data analytics, statistics, hardware and software engineering, linguistics, neuroscience, philosophy, and psychology. For operational tasks including data analysis, prediction, object classification, natural language processing, and intelligent data retrieval, artificial intelligence primarily uses machine learning and deep learning. It's crucial to remember that the majority of AI systems in use today are classified as "narrow AI." This indicates that rather than displaying broad human-like intelligence, they are made to carry out certain tasks dependent on their programming and training.

Historical Context

The creation of thinking machines has conceptual origins in ancient philosophy and folk tales, envisioning "automatons" that moved independently. However, AI was formally established as an academic discipline in 1956 at the Dartmouth Conference, a landmark event where John McCarthy coined the term "artificial intelligence." This pivotal gathering brought together researchers who shared the conviction that any aspect of learning or intelligence could, in principle, be precisely described and simulated by a machine [25].

Following this foundational period, AI research saw periods of significant optimism and breakthroughs, such as the development of Shakey the Robot (1966-1972) at the Stanford Research Initiative, which could navigate environments using sensors and a TV camera [2]. However, the field also experienced "AI winters"–periods of disappointment and funding cuts, notably after a critical report in 1974 by Sir James Lighthill, which claimed researchers had over-promised and under-delivered on AI's potential. Despite these challenges, AI continued to evolve, particularly in the 21st century, driven by advancements in computing power and algorithms, leading to significant progress in areas like symbolic logic, machine learning, and neural networks [25].

Core Components and Capabilities

AI encompasses a variety of toolsets and methodologies, enabling systems to tackle complex problems that are challenging for traditional algorithmic approaches . Key components and capabilities include:

Knowledge Representation: This involves structuring and storing information in a way that a computer system can utilize for complex tasks. Ontologies, for example, structure knowledge as interconnected concepts within a specific domain. This organizational approach enables AI programs to respond to inquiries and derive conclusions about factual real-world information.

Reasoning and Problem-Solving: Initial AI research concentrated on creating algorithms that emulated the sequential thought processes humans use to solve puzzles or perform logical deductions. AI systems employ reasoning to infer new information and make decisions based on the available data.

Learning: This is a crucial aspect where AI systems enhance their performance progressively by processing extensive datasets. They identify patterns and relationships that might be overlooked by human observation. This capability is primarily driven by machine learning techniques.

Natural Language Processing (NLP): This specialized branch allows machines to comprehend, interpret, and generate human language, whether in spoken or written form.

Perception: AI systems possess the ability to interpret visual information (through Computer Vision) and other forms of sensory data from their environment. This includes capabilities such as speech recognition, image classification, and object recognition.

Robotics and Actuation: AI can be seamlessly integrated with physical systems, empowering robots to execute tasks autonomously or semi-autonomously within the physical world.

Branches of Artificial Intelligence

AI covers a wide range of specialized fields, each of which focuses on different aspects of intelligence:

ML or machine learning: ML is a fundamental branch of AI that focuses on leveraging large datasets to train algorithms. Without the need for explicit programming, this method allows them to identify patterns and then make predictions or conclusions. In essence, machine learning (ML) enables algorithms to learn directly from data, improving their performance over time.

Supervised Learning Algorithms

A range of algorithms are used in the field of supervised learning, selected according to the properties of the data and the goal of the prediction. Important instances consist of:

By fitting a straight line (or a hyperplane in higher dimensions) to one or more independent variables, the statistical technique known as linear regression is used to predict a continuous dependent variable. It works by reducing the sum of the squared deviations between the output values that were expected and those that were actually obtained. For predicting and comprehending variable interactions, linear regression models are prized for their ease of use, interpretability, and wide range of applications in domains including business, behavioral, environmental, biological, and social sciences [26].

Although *Logistic Regression* and linear regression share fundamental similarities, logistic regression is designed for binary classification issues with dichotomous outcome variables (e.g., predicting the existence or absence of an event). Instead of predicting a continuous value directly, it uses a sigmoid function to represent the likelihood of a particular result. In this way, a linear combination of independent variables is efficiently converted into a probability score between 0 and 1. By examining all variable relationships at once, this approach has the benefit of reducing confounding effects [27].

Strong supervised machine learning methods, *Support Vector Machines (SVMs)* are primarily utilized for classification tasks, while they can also be employed for regression. Finding the ideal hyperplane in a high-dimensional feature space that offers the greatest separation between data points from various classes is the basic idea behind support vector machines (SVM). To improve the model's capacity for generalization and robustness to noisy data, the "margin" the distance between this hyperplane and the nearest data points (also known as support vectors) from each class is maximized. SVMs use "kernel functions" to implicitly project data into a higher-dimensional space where a linear division is possible in cases when the data cannot be divided linearly.

Decision Trees: Suitable for both classification and regression, decision trees are a non-parametric supervised learning approach. A root node, internal nodes (decision nodes), and leaf nodes (terminal nodes) make up their hierarchical, tree-like structure. A feature or trait is assessed at each internal node in order to divide the data into homogeneous subsets, with branches signifying possible outcomes. The categorization or prediction criteria are embodied in the paths leading from the root to the leaf nodes. Their broad appeal is a result of their interpretability and simplicity of visualization.

By building a "forest" out of several uncorrelated decision trees during the training stage, *Random Forests* is an ensemble learning technique that outperforms individual decision trees. Using a technique called bagging in conjunction with randomized node optimization, each tree in the forest is trained on a unique, random subset of the data and characteristics. A majority vote among the trees determines the final forecast for classification, whereas the average of the trees' predictions determines the final prediction for regression. Predictive accuracy is increased and the potential of overfitting, a major problem with single decision trees, is greatly decreased with this ensemble approach. Additionally, Random Forests inherently reveal the significance of variables. [28]

Gradient Boosting Machines (GBMs): GBMs are an extremely powerful ensemble learning method that builds a strong predictive model by successively combining several "weak" prediction models, usually shallow decision trees. Boosting algorithms teach models iteratively as opposed to bagging techniques (like Random Forests), which train models in parallel. The specific goal of every new model is to fix the mistakes (residuals) made by its forerunners. The procedure iteratively fits new models to the loss's negative gradient by utilizing the concepts of gradient descent to minimize a predetermined loss function. GBMs can grasp intricate non-linear correlations in data and achieve remarkable predictive accuracy in both regression and classification applications thanks to this repeated error correction.

Unsupervised Learning Techniques

A data mining technique called clustering divides unlabeled data points into groups according to their innate similarities or differences. The algorithms search through raw, unclassified datasets for underlying structures or patterns.

K-*means Clustering:* One of the most widely used clustering methods, K-means is an exclusive, or "hard," clustering algorithm, meaning each data point is assigned to precisely one clusterA predetermined number (k) of related groups are created from a dataset using this iterative, centroid-based approach. The sum of squared distances between data points and their designated cluster centroids is minimized in order to do this. Each observation is iteratively assigned to its nearest centroid by the method, which usually starts by initializing k centroids (usually at random). Centroids are then recalculated based on these new cluster assignments until convergence is achieved.

The density-based clustering technique known as *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* finds clusters by separating dense regions of data points by less dense areas [28, 29]. In contrast to K-means, DBSCAN may find clusters of any shape and does not require the number of clusters to be predetermined [28, 29]. Data points are categorized into three groups: noise points (outliers that do not belong to any cluster), border points (placed close to core points but without enough neighbors to be core themselves), and core points (those with enough neighbors within a given radius). DBSCAN works very well with outliers and noisy datasets.

A technique called "*dimensionality reduction*" seeks to reduce a dataset's features (dimensions) while retaining as much pertinent information as feasible. Complex datasets are simplified by this method, which facilitates visualization and analysis, particularly in high-dimensional spaces [29].

Principal Component Analysis (PCA) is a popular technique for reducing linear dimensionality. It converts a big collection of correlated variables into principle components, which are smaller sets of uncorrelated variables. The greatest variance in the data is captured by these principal components, which are linear combinations of the original variables. The initial main components can be utilized in later studies without worrying about multicollinearity because they usually represent the biggest percentage of the total variance. Because PCA is sensitive to the scale of variables, it is necessary to standardize or normalize the data as a preprocessing step.

The non-linear dimensionality reduction method known as *t-Distributed Stochastic Neighbor Embedding (t-SNE)* is mostly used to visualize high-dimensional data in two or three dimensions. In order to reduce the divergence between these probabilities and those in the low-dimensional embedding, it first transforms similarities between data points in the high-dimensional space into joint probabilities. When it comes to maintaining local structures in the data, t-SNE is very good at keeping dissimilar data points apart and similar data points near together in the lower-dimensional representation. It has been widely used to extract intuitive insights from complicated information in domains including as natural language processing, neuroscience, and genomics.

Finding observations, occurrences, or data points that substantially depart from the usual, standard, or anticipated behavior within a dataset and are therefore inconsistent with the rest of the data is known as anomaly detection (outlier detection). [30]. Anomalies can arise from various causes, such as errors, noise, or malicious activity, and their identification is crucial for proactive decision-making and incident response across industries like finance, healthcare, and cybersecurity [30, 31]. Anomaly detection systems can uncover both unintentional anomalies (due to errors or noise) and intentional anomalies (due to specific events or actions). A range of methods, including statistical tests and machine learning algorithms (such as Random Forest and Autoencoder), are utilized for this task, often requiring appropriate data preprocessing to enhance accuracy [31].

Reinforcement Learning Concepts and Algorithms

Markov Decision Processes (MDPs) are often used to model basic reinforcement learning ideas. In situations where outcomes are partially random and partially under the decision-maker's control, MDPs offer a mathematical framework for sequential decision-making. [32] Formally, an MDP is defined by:

A collection of states (S), representing the environment's configuration and the agent's position.

A set of actions (A) available for the agent to take.

Transition probabilities (P(s’|s,a)), which specify the likelihood of moving to a new state (s′) from state (s) after performing action (a).

A reward function (R(s,a,s′)), which quantifies the immediate reward received upon a state transition.

The primary objective within an MDP is for the agent to learn an optimal policy a mapping from states to actions that maximizes the expected cumulative reward over time.

An explicit model of the dynamics of the environment is not necessary for an agent to find the best course of action thanks to the model-free reinforcement learning algorithm known as Q-learning. It estimates the "Q-value" for every state-action pair iteratively. The predicted future reward for performing a specific action in a state and then following an ideal policy is indicated by this Q-value. The Bellman equation, which combines the greatest Q-value of the subsequent state with the immediate reward, is the fundamental mechanism of the algorithm for updating these Q-values. Although tables are frequently used in basic Q-learning to store Q-values, this method is not feasible in settings with extremely vast state-action spaces.

Deep Q-Networks (DQNs) extend the concept of Q-learning by incorporating deep neural networks to approximate the Q-value function. [46] This innovation overcomes the scalability limitations of traditional Q-learning, especially in environments characterized by high-dimensional state spaces (raw pixel data from video games) [33, 34]. DQNs have achieved remarkable successes, notably surpassing human performance in complex Atari 2600 games. Key architectural elements of DQNs frequently include a convolutional neural network (CNN) acting as a value approximator, alongside the use of a distinct "target network" during training to stabilize the learning process [33].

One type of reinforcement learning technique that directly optimizes a parameterized policy is called a policy gradient method. Gradient descent is used to optimize this process in relation to the expected return, or long-term cumulative reward. These approaches directly learn a policy that maps states to actions, which can be either stochastic or deterministic, as opposed to learning value functions, as is the case with Q-learning. Policy gradient algorithms are particularly appealing due to their capacity to manage continuous state and action spaces and their guarantee of converging to at least a locally optimal policy. They intrinsically learn stochastic policies, which can facilitate exploration and lead to more gradual policy adjustments during learning [35, 36].

Actor-critic methods synergize the strengths of both policy-based ("actor") and value-based ("critic") reinforcement learning approaches. The "actor" component is responsible for selecting actions based on the current policy, while the "critic" component estimates the value of executing those actions (i.e., the value function). The critic provides feedback to the actor by evaluating the chosen actions, and this evaluation is then used to update the actor's policy through policy gradients. This hybrid methodology frequently results in more efficient and stable learning, especially in complex decision-making and control tasks, and can manage continuous action spaces more effectively than pure Q-learning.

A more complex subset of machine learning called deep learning (DL) uses multi-layered artificial neural networks. These networks, which draw inspiration from the structure of the human brain, are skilled at finding complex patterns and connections in data, which has resulted in notable advancements in fields like natural language processing and picture identification.

Neural Networks

The fundamental architecture of deep learning is neural networks, particularly deep neural networks. Their architecture, which consists of interconnected layers of artificial neurons processing input to produce a desired output, is modeled after the composition and operation of the human brain [37, 40].

Multilayer Perceptrons (MLPs): MLPs represent the most widely recognized artificial neural network architecture. They are defined by their organization of neurons into multiple distinct layers: an input layer, one or more hidden layers, and an output layer. Neurons within a specific layer do not interact with each other; instead, they transmit their signals forward, creating a feedforward structure. MLPs possess the capability to approximate any continuous, bounded, differentiable, non-linear function with arbitrary precision, making them highly adaptable for tasks such as pattern classification, data mining, and function approximation.

Activation Functions: These are essential components within neural networks that shape the outputs of individual artificial neurons. An activation function introduces non-linearity into the network, enabling it to learn intricate patterns and relationships that purely linear models cannot capture. Frequently used activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh [38, 40].

A fundamental procedure used to train neural networks, especially MLPs, is backpropagation. It works by calculating the gradient of the loss function with respect to the weights and biases of the network. The network can then iteratively modify its weights and biases to reduce the overall error by propagating the mistake backward from the output layer through the hidden layers [38, 40].

Optimisation Algorithms: These techniques modify a neural network's weights and biases during training in order to reduce the loss function. Gradient Descent (and its variations, such as Stochastic Gradient Descent and Mini-batch Gradient Descent), Adam, RMSprop, and AdaGrad are examples of popular optimization methods. They control the network's learning process and convergence to the best answer [38, 40].

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent a specialized category of neural networks that excel at processing structured array data, most notably images. They are extensively applied in tasks such as image classification, object detection, segmentation, and video processing. [41]

Convolutional Layers: These layers form the fundamental building blocks of CNNs. They utilize learnable "kernel filters" that traverse the input data (e.g., an image) to perform convolution operations. This process extracts essential features like edges, textures, or recurring patterns. A convolutional layer's configuration is determined by its kernel size, stride length (the number of steps the kernel advances), and padding (the addition of zeros around the input boundaries) [41].

Pooling Layers: Used to decrease the spatial dimensions of the feature maps, pooling layers usually come after convolutional layers. The overall computing load and the number of parameters are successfully reduced by this action. Additionally, this downsampling strategy helps to mitigate overfitting and strengthens the model's resistance to small input fluctuations. Max-pooling, which chooses the largest value within a specified region, and average-pooling, which determines the average value within a region, are examples of common pooling processes.

Architectures

Over the years, several groundbreaking Convolutional Neural Network (CNN) architectures have been developed, each contributing significantly to the field:

LeNet-5: Introduced in 1998, LeNet-5 stands as one of the pioneering CNNs, gaining widespread recognition for its success in handwritten digit recognition. Its typical structure includes two convolutional layers followed by three fully connected layers [42].

AlexNet: Debuting in 2012, AlexNet was a considerably deeper and larger CNN that dramatically enhanced image classification performance. It was among the first architectures to extensively employ Rectified Linear Units (ReLUs) as activation functions and to harness the power of GPUs for accelerated training. This network features five convolutional layers and three fully connected layers [42].

VGG (Visual Geometry Group): VGG networks are notable for their straightforward design and considerable depth, primarily utilizing stacked 3x3 convolutional filters across multiple layers. Common iterations, such as VGG-16 and VGG-19, refer to their respective depths of 16 and 19 layers.

ResNet (Residual Network): Introduced in 2015, ResNet resolved the critical vanishing gradient problem prevalent in very deep neural networks. It achieved this by incorporating "residual connections" or "skip connections," which enable gradients to flow directly through layers. This innovation made it possible to train networks with hundreds or even thousands of layers, leading to substantial performance improvements [42].

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a type of neural network built to handle sequential data. What makes them unique is that the output from a previous step feeds directly into the current step, making them perfect for tasks like time series analysis, natural language processing, and speech recognition.

A specific type of RNN called *Long Short-Term Memory (LSTM)* was created to address the vanishing gradient issue, which frequently prevents conventional RNNs from picking up long-term patterns. Three "gates" (input, output, and forget gates) and a cell state that retains data over extended periods of time are common features of an LSTM unit. The network may choose recall or forget significant data in lengthy sequences thanks to these gates, which regulate the flow of information into and out of the cell. Time series analysis, machine translation, and speech recognition all make extensive use of LSTMs.

The vanishing gradient problem is also addressed with Gated Recurrent Units (GRUs), a more straightforward variant of LSTMs that were first introduced in 2014. With less parameters, they function similarly to LSTMs. To control information flow and update the hidden state selectively, GRUs employ two gates: an update gate and a reset gate. They are frequently used in time series prediction and natural language processing (NLP) [43].

Sequence-to-Sequence Models (Seq2seq): Seq2seq models are a family of machine learning approaches designed to convert one sequence into another. They're frequently used in natural language processing for tasks like language translation, text summarization, and building conversational AI. A typical Seq2seq model consists of two neural networks: an encoder network that turns an input sequence into a numerical "context vector," and a decoder network that then transforms this vector into an output sequence. An "Attention mechanism" is often added to improve performance by letting the decoder focus on the most relevant parts of the input sequence.

Generative Models

Generative models are a category of machine learning models engineered to produce new data instances that closely resemble their training data.

Variational Autoencoders (VAEs): VAEs are a type of generative model rooted in deep learning. They learn a compressed representation of input data within a "latent space." Unlike conventional autoencoders, VAEs model this latent space probabilistically, learning the parameters (mean and variance) of a specific probability distribution (e.g., Gaussian). The decoder then samples from this distribution to reconstruct the original input, allowing VAEs to generate novel, similar data instances. They prove valuable for tasks such as data generation and creating low-dimensional representations of target data.

GANs or Generative Adversarial Networks: A "generator" and a "discriminator" are two competing neural networks that make up the potent family of generative models known as GANs. While the discriminator tries to distinguish between real data from the training set and artificial data generated by the generator, the generator is entrusted with producing new data instances (such as images) from random noise. Both networks advance as a result of this adversarial process: the discriminator becomes better at spotting phony data, and the generator tries to create more convincing data to trick the discriminator. GANs have achieved remarkable success in generating realistic images, videos, and other data types [44].

Other Key AI Branches

The field of *Natural Language Processing (NLP)* focuses on how computers and human language interact, allowing machines to understand, interpret, and produce text and voice that sounds human.

Computer Vision: This field makes it possible for machines to decipher and comprehend visual data from pictures and videos, which makes jobs like object detection, facial recognition, and image classification easier.

Expert Systems: Made to mimic human decision-making, these systems use a body of rules and information to handle complicated issues, usually in specialized fields.

Robotics: This branch integrates AI principles to develop and design robots capable of performing tasks autonomously or semi-autonomously in the physical world.

Fuzzy Logic: Employed for reasoning with inherently uncertain or imprecise concepts, fuzzy logic offers a flexible framework for implementing machine learning techniques.

Evaluation and Validation

In the development lifecycle of machine learning models, robust evaluation and validation are indispensable steps. These processes are crucial for ensuring that models consistently perform reliably and exhibit strong generalization capabilities when confronted with unseen data. Furthermore, they facilitate a comprehensive understanding of a model's strengths and limitations, actively help in averting problems like overfitting, and guide the selection of the most fitting model for a specific task.

Data Splitting Strategies

To effectively assess a machine learning model, a dataset is typically divided into three distinct subsets:

A dataset is usually split into three separate groups in order to evaluate a machine learning model:

Training Set: Specifically used to train the model, this comprises the majority of the data. In this stage, the model iteratively modifies its internal parameters to discover underlying patterns and relationships.

The validation set, also known as the development set or dev set, is a subset that provides an objective assessment of a model's performance throughout the training process. It is crucial for choosing the best model from a group of candidates and for fine-tuning the model's hyperparameters, which are settings made before training, like the number of layers in a neural network. Overfitting, a situation in which a model performs well on its training data but badly on new, unknown data, is actively avoided by using a separate validation set [45].

Test Set: This set is kept completely apart from the validation and training procedures. It serves as an accurate gauge of the model's generalization ability and offers an objective evaluation of the final, chosen model's performance on really unseen data. To guarantee a correct evaluation, all three sets should ideally follow the same probability distribution.

A fundamental method used to assess model performance and identify ideal tuning parameters is cross-validation (CV), which is particularly useful when data is limited or more consistent outcomes are desired. In this procedure, the data is divided into several subsets, and the model is iteratively trained on some of the subsets while being validated on the others. This approach reduces the risk of overfitting that comes with a single train-validation split while also assisting in measuring the model's prediction performance and generalizability to unseen data. k-Fold cross-validation is a popular cross-validation method in which the dataset is split into k folds of equal size. Following that, the model is trained k times, utilizing a different fold for validation and the remaining k−1 folds for training in each iteration [46].

Model Selection and Hyperparameter Tuning

Model Selection is the process of choosing the most appropriate machine learning model or algorithm for a particular predictive modeling task [46]. This often necessitates comparing the performance of various candidate models on the validation set, while also considering aspects such as accuracy, computational complexity, and interpretability.

Hyperparameter Tuning involves optimizing the external configurations of a model known as hyperparameters which are defined before the training process commences (learning rate, the number of trees in a random forest, or regularization strength). These parameters significantly influence how effectively a model performs on unseen data. Hyperparameter tuning is the systematic exploration for the optimal combination of these hyperparameters to maximize a model's performance. Common strategies employed include grid search (which tests every possible combination within a pre-defined range) and random search (which samples combinations randomly from the search space) [45].

Performance Metrics

Performance metrics serve as quantitative benchmarks for evaluating the effectiveness and precision of machine learning models. The selection of an appropriate metric is highly dependent on the particular task (e.g., classification, regression) and the overarching business objectives.

For Classification Tasks

Accuracy: The percentage of accurately predicted instances compared to all instances is measured by this metric. It is especially useful for datasets that are balanced and give equal weight to each type.

Precision measures the accuracy of the model's positive predictions by displaying the ratio of true positive predictions to the total number of positive predictions (sum of true positives and false positives).

Recall (Sensitivity): This measure determines the proportion of actual positive cases (false negatives plus true positives) to true positive forecasts. It accurately assesses the model's ability to find all pertinent positive cases.

F1-Score: This balanced performance metric, which is particularly useful for handling unequal class distributions, is calculated as the harmonic mean of precision and recall.

Confusion Matrix: This tabular representation provides a concise overview of the performance of a classification model. The numbers of false positives, false negatives, true positives, and true negatives are shown.

ROC Binary classifier performance is assessed using the Curve and AUC (Area Under the Receiver Operating Characteristic Curve) across a range of threshold values. AUC offers a single scalar value that summarizes the classifier's overall performance.

Cross-Entropy Loss, or log loss, is a statistic used to assess how well a classification model performs when its predictions are represented as probability values between 0 and 1.

For Regression Tasks

Model performance for regression tasks is assessed using a number of important metrics:

The average of the squared discrepancies between the expected and actual values is determined by the Mean Squared Error (MSE) statistic. Larger mistakes are severely penalized.

The root mean squared error (RMSE), which provides an error measure in the same units as the dependent variable, is just the square root of the MSE [59].

The average of the absolute discrepancies between the expected and actual values is represented by the Mean Absolute Error (MAE) measure. In contrast to MSE, it is less susceptible to outliers.

R-squared: This figure represents the percentage of the dependent variable's variance that can be anticipated based on the independent factors. In essence, it displays how well the model accounts for the data that was seen [47].

The average of the absolute % mistakes is measured by the Mean Absolute % Error, or MAPE.

These criteria are essential for creating AI models that are not just strong and dependable but also successful at extrapolating to real-world situations, when paired with suitable data segmentation and validation techniques.

Understanding the Confusion Matrix in Machine Learning

The confusion matrix stands as a fundamental instrument for assessing the performance of classification models, applicable to both binary and multi-class classification challenges [48]. It offers a detailed breakdown of a model's correct and incorrect predictions, providing insights that go beyond simple accuracy scores.

Components of a Confusion Matrix: For a binary classification scenario, a confusion matrix is structured as a 2×2 table, illustrating four crucial outcomes by comparing the model's predicted labels against the true labels:

True Positives (TP): Instances where the actual label is positive, and the model accurately identified it as positive. This signifies a correct detection of the positive class.

True Negatives (TN): Instances where the actual label is negative, and the model correctly predicted it as negative. This indicates a precise identification of the negative class.

False Positives (FP) / Type I Error: Instances where the actual label is negative, yet the model erroneously predicted it as positive. These are frequently termed "false alarms."

False Negatives (FN) / Type II Error: Instances where the actual label is positive, but the model mistakenly predicted it as negative. These represent "missed cases."

Metrics Derived from Confusion Matrix Data: The values within a confusion matrix are used to compute various performance metrics, each offering a distinct perspective on the model's effectiveness: [48]

Accuracy: Defined as $\frac{TP+TN}{TP+TN+FP+FN}$, accuracy denotes the proportion of all classifications that were correct. While intuitively appealing, it can be misleading, especially for imbalanced datasets where one class significantly outnumbers the other.

Precision: Calculated as $\frac{TP}{TP+FP}$, precision measures the proportion of positive predictions that were genuinely correct. It is particularly vital when the cost associated with false positives is high (e.g., incorrectly flagging a legitimate transaction as fraudulent).

Recall (Sensitivity or True Positive Rate - TPR): Determined by $\frac{TP}{TP+FN}$, recall assesses the proportion of actual positive instances that the model successfully identified. It gains importance when the cost of false negatives is high (e.g., failing to detect a critical disease). Recall is also the complement of the Type II error rate.

F1-Score: The harmonic mean of precision and recall, expressed as $\mathrm{F1} = 2 \times \frac{(\mathrm{Precision} \times \mathrm{Recall})}{(\mathrm{Precision} + \mathrm{Recall})}$. It provides a balanced measure, proving especially useful for datasets with imbalanced class distributions where accuracy might be deceptive.

Specificity (True Negative Rate - TNR): Calculated as $\frac{TN}{FP+TN}$, specificity measures the proportion of actual negative instances that were correctly identified.

Type I Error (False Positive Rate - FPR): Defined as $\frac{FP}{TN+FP}$, this represents the rate at which the model incorrectly predicts the positive class when the actual class is negative.

Type II Error (False Negative Rate - FNR): Defined as $\frac{FN}{FN+FP}$, this is the rate at which the model erroneously predicts the negative class when the actual class is positive.

Confusion Matrix for Binary Classification: As elaborated above, for binary classification scenarios, the confusion matrix is a 2×2 table that clearly delineates the four outcomes (TP, TN, FP, FN) for two classes (e.g., positive/negative, spam/not spam).

Confusion Matrix for Multi-class Classification: When a model deals with more than two classes, the confusion matrix expands into an N×N matrix, where N represents the number of classes [74]. Typically, each row signifies the instances belonging to an actual class, while each column represents instances assigned to a predicted class. The elements along the main diagonal indicate correct classifications for each respective class, whereas off-diagonal elements illustrate misclassifications between different classes. Metrics such as precision, recall, and F1-score can then be computed for each class individually (e.g., using micro-average, macro-average, or weighted average methods) [48].

Linear Models: Regression and Classification

Linear models constitute a foundational category of statistical models employed in machine learning for both regression and classification problems. Their defining characteristic is the assumption that the relationship between the input features and the output variable can be effectively represented by a linear function.

As was previously mentioned, *Linear Regression* is a statistical technique designed especially for regression problems in which predicting a continuous numerical output variable is the goal. By fitting a linear equation to the observed data, it determines the connection between a dependent variable and one or more independent variables. Finding the best straight line (or hyperplane in higher dimensions) that minimizes the sum of squared residuals, or the differences between expected and actual values, is the goal of this approach. While multiple linear regression includes several independent variables, simple linear regression just uses one. Because of their simplicity, interpretability, and strong theoretical foundations, linear regression models are widely used in a variety of scientific and commercial domains to clarify variable interactions and generate predictions [25, 48].

Linear Model for Classification (Logistic Regression): Despite its name, which includes "regression," logistic regression is fundamentally a classification algorithm. It leverages a linear model to predict the probability of a binary outcome (e.g., 0 or 1, true or false). Instead of directly predicting a continuous value, it applies a sigmoid (logistic) function to the linear combination of input features. This sigmoid function compresses the output of the linear equation into a probability value ranging from 0 to 1, which can then be subjected to a threshold to assign a class label. Logistic regression is especially valuable when the dependent variable is dichotomous, allowing for the evaluation of the influence of multiple explanatory variables on the odds of an event occurring. It serves as a potent tool for analyzing associations between variables and generating predictions in classification contexts.

CHAPTER 3. SIMULATION, IMPLEMENTATION AND RESULTS

3.1 Overview of the Power System Datasets

The Power System Datasets are designed to simulate various operational and attack scenarios in electric transmission systems, providing a comprehensive resource for research in cyber-attack detection within SCADA systems. Each data sample comprises 128 features, including synchrophasor measurements and logs from Snort, simulated control panels, and relays. The datasets are categorized into three classification schemes:

1. *Binary Classification:* Differentiates between normal operations and attacks.
2. *Ternary Classification:* Includes normal operations, attacks, and no-operation states.
3. *Multi-class Classification:* Each of the 37 event scenarios is treated as a distinct class.

The datasets encompass a variety of scenarios to reflect real-world conditions:

- *Normal Operations:* Standard functioning of the power system without disturbances.
- *Natural Events:* Such as short-circuit faults and line maintenance activities.
- *Cyber Attacks:*
  - *Remote Tripping Command Injection:* Unauthorized commands sent remotely to trip circuit breakers.
  - *Relay Setting Change:* Alteration of relay settings to disrupt normal operations.

- *Data Injection:* Manipulation of measurement data to mislead system monitoring and control.

These scenarios are instrumental in training and evaluating machine learning models for detecting and classifying cyber threats in power systems.

3.2 Accessing the Datasets

The Power System Datasets, along with detailed documentation, can be accessed through the following link: [ICS Cyber Attack Datasets by Tommy Morris]

For researchers and practitioners trying to improve the cybersecurity of SCADA systems in power networks, this collection offers extensive materials. Research on Cyber Attack Detection for Industrial Power Control Systems (ICS).

The test system employed, as well as the various natural and artificial situations, are detailed together with machine learning classification algorithms for power system disturbance discrimination.

Scenario Types:

1. Short-circuit fault: This is a power cable short circuit that can occur anywhere along the cable; the range of percentages indicates the location.

2. Line maintenance: The remote relay trip command trips a breaker or breakers for maintenance.

3. The assault known as remote tripping command injection (assault) occurs when a relay receives a command that causes a breaker to open. It can only be executed after the defenses have been penetrated by an outside attacker.

4.Relay setting change (Attack) – relays are configured with a distance protection scheme and the attacker changes the setting to disable the relay operation so that relay will not trip for a valid fault or a valid command.

5.Data Injection (Attack) – in this we simulate a legitimate fault by modifying values to parameters like current, voltage, sequence components etc., to blind the operator and result in a black out.

The events were classified according to three classification schemes:

- Multiclass - All 37 event scenarios, including attack events, natural events, and normal operations, were separate classes and were predicted separately by the learners
- Three-class - The 37 event scenarios were assigned to 3 classes: attack events (28 events), natural event (8 events) or "No events" (1 event).
- Binary - The 37 event scenarios were labeled as an attack (28 events) or normal operations (9 events). The information was drawn from 15 data sets that had thousands of individual samples of measurements throughout the power system for both types of events Datasets, which we are referring to were randomly sampled at 1% to reduce the size and evaluate the effectiveness of small sample sizes. For our experiment, we used an average of 294 "No event" examples, 3,711 attack examples and 1,221 natural events examples throughout the classification schemes. Time and date features were excluded since scenarios were executed chronologically and time and date would be optimal in classifying the data. For all three schemes, Multiclass, Three-class and Binary, we experimented with Adaboost+JRipper learners on 15 datasets. When conducting the experiments we opted to apply the tenfold or 10x cross validation approach. When testing with this approach we divided the dataset into 10 folds by randomly taking instances from each class. We trained the model on a ninety percent subset of the data and tested it on the remaining ten percent of the data so as to establish the learner's performance. We did this for all learners and all datasets and then averaged across the fifteen datasets to summarize the results.The codes involved are given in Appendix.

To give an example of an algoritm we take the following example. We will use the K-nearest Neighbors algorithm for classification.

For example, we have a classification dataset of viruses

Table 3.1. Dataset for Classification of Viruses

ID	System calls	Encryption	Disk write	API Access	Network activity	Size (KB)	Class
1	1200	0	10	0	0	450	0 (Benign)
2	3400	1	80	1	1	900	1 (Trojan)
3	5000	1	150	1	1	1200	2 (Ransomware)
4	1500	0	5	0	0	370	0
5	4200	1	130	1	1	110	2
6	3100	1	60	1	1	750	1
7	1100	0	7	0	0	410	0
8	4700	1	120	1	1	1000	2
9	3000	1	50	1	1	670	1
10	1400	0	6	0	0	395	0

Encryption attribute means does the program use cryptographic methods or not.

Benign (0) usually doesn't encrypt data or uses it securely

Trojan (1) may use weak encryption

Ransomware (2) actively uses encryption to encrypt the victim's files

Disk write attribute reflects how actively the program interacts with the system. It can be the total number of write operations, their frequency...

Benign (0) writes little

Trojan (1) can write malicious components

Ransomware (2) writes a lot

< 10 → Benign, ≈ 80 → Trojan, > 150 → Rams.

System calls is a way for a program to interact with the operating system kernel, for example, total number of system calls during some period.

Benign (0) usually calls standard calls

Trojan (1) calls suspicious system calls

Ransomware (2) generates a lot of calls

Table 3.2. Means and Deviation of Program's Cryptographic methods

ID	**System calls**	**Encryption**	**Disk write**	**API access**	**network actives**	**Size (KB)**	**Class**
1	1200	0	10	0	0	450	0 (Benign)
2	3400	1	80	1	1	900	1 (Trojan)
3	5000	1	150	1	1	1200	2 (Ransom)
4	1500	0	5	0	0	370	0 (Benign)
5	4200	1	130	1	1	110	2 (Ransom)
6	3100	1	60	1	1	750	1 (Trojan)
7	1100	0	7	0	0	410	0 (Benign)
8	4700	1	120	1	1	1000	2 (Ransom)
9	3000	1	50	1	1	670	1 (Trojan)
10	1400	0	6	0	0	395	0 (Benign)
mean	2860	0.6	61.8	0.6	0.6	724.5	
deviation	1489.36974	0.516398	56.09278	0.516398	0.516398	313.7449	

Table 3.3. Standardtized Matrix

ID	**System calls**	**Encryption**	**Disk write**	**API access**	**network actives**	**Size (KB)**	**Class**
1	-1.11456541	-1.1619	-0.92347	-1.1619	-1.1619	-0.87491	0 (Benign)
2	0.36255947	0.774597	0.324465	0.774597	0.774597	0.559372	1 (Trojan)
3	1.43684868	0.774597	1.572395	0.774597	0.774597	1.516242	2 (Ransom)
4	-0.91313793	-1.1619	-1.01261	-1.1619	-1.1619	-1.1299	0 (Benign)
5	0.89970367	0.774597	1.215843	0.774597	0.774597	-1.96832	2 (Ransom)
6	0.161141997	0.774597	-0.03206	0.774597	0.774597	0.080576	1 (Trojan)
7	-1.18170791	-1.1619	-0.97693	-1.1619	-1.1619	-0.90241	0 (Benign)
8	1.235421904	0.774597	1.037567	0.774597	0.774597	0.878102	2 (Ransom)
9	0.093999253	0.774597	-0.21037	0.774597	0.774597	-0.17371	1 (Trojan)
10	-0.98028042	-1.1619	-0.99476	-1.1619	-1.1619	-1.05022	0 (Benign)

a) We need the standardized dataset:

$$X_{new} = \frac{X-\mu}{\vartheta} \qquad (3.1)$$

μ - mean for each feature

σ - standard deviation = $\sqrt{\frac{\sum(X-\overline{X})}{n-1}}$ (3.2)

Then we determine a covariance matrix. As the dataset is normalized then in an equation

$\mathrm{cov}(\mathrm{x},\mathrm{y}) = \frac{1}{\mathrm{n}-1}$ (3.3)

$\sum_{i=1}^{n} (X_i - \overline{X}) \cdot (Y_i - \overline{Y})$ (3.4)

The mean $\overline{X} = 0, \overline{Y} = 0$ and $\mathrm{cov}(\mathrm{X},\mathrm{Y}) = \frac{1}{\mathrm{n}-1}$

$\sum_{i=1}^{n} X_i Y_i$ (3.5)

We have dataset $\epsilon \mathrm{R}^{10\times 6}$ Then our covariance matrix will be $\epsilon \mathrm{R}^{6\times 6}$

and will be calculated as $\mathrm{cov.m} = \frac{1}{\mathrm{n}-1} \mathrm{X}^{\mathrm{T}} \cdot \mathrm{X}$

X^T is the transposed normalized matrix.

For example:

covariance between system calls and disk write will be:

$$cov_{1,2} = \frac{1}{9}((-1.115)(-0.923) + (0.363)(0.324) + \ldots = 0.979)$$

Take into account that $\mathrm{cov}(\mathrm{X},\mathrm{X}) = 1$

Table 3.4.Covariance Matrix

	System calls	**Encryption**	**Disk write**	**API Access**	**Network act.**	**Size**
System calls	1	0.901	0.979	0.901	0.901	0.973
Encryption	0.901	1	0.841	1	1	0.873
Disk write	0.979	0.841	1	0.841	0.841	0.993
API Access	0.901	1	0.841	1	1	0.873
Network activity	0.901	1	0.841	1	1	0.873
Size	0.973	0.873	0.993	0.873	0.841	1

All covariances are high. It means strong dependency between features.

Eigenvalues

$\mathrm{cov} \cdot \mathrm{w} = \lambda \cdot \mathrm{w}$ (3.6)

$\det(\mathrm{cov} - \lambda \mathrm{I}) = 0$ (3.7)

I - unit matrix 6×6

For covariance matrix trace of the matrix (sum of diagonal values)

For our matrix

$T_2(cov) = 1 + 1 + 1 + 1 + 1 + 1 = 6$

$\sum {\lambda_i}^2 = \sum_{i,j} {cov_{ij}}^2$ (3.8)

Approximated eigenvalue (Power method)

$\lambda_{max} \approx \frac{\vartheta^T \cdot \sum cov_i}{\vartheta^T \cdot \vartheta}$ (3.8)

where σ - random vector

For example: $\vartheta = [1\ 1\ 1\ 1\ 1\ 1]^{\mathrm{T}}$

Then $\sum cov$ - sum of rows

Example for the first row:

$[1\ 0.901\ 0.979\ 0.901\ 0.901\ 0.973] \cdot [1\ 1\ 1\ 1\ 1\ 1]^{\mathrm{T}}$

For all rows we will have the following

$$\vartheta^T \cdot cov_i = \begin{matrix} 5.655 \\ 5.616 \\ 5.635 \\ 5.616 \\ 5.616 \\ 5.693 \end{matrix}$$

Then $\vartheta^{T \sum cov_i} = sum\ of\ rows = 33.83/\vartheta^T\vartheta = 6$

$$\lambda_{max} \approx \frac{33.83}{6} \approx 5.638$$

In Matlab we determined the following eigenvalues:

$$\lambda_1 = 5.5971 \rightarrow 93.3\% = \frac{5.5971}{6}$$

$$\lambda_2 = 0.3759$$

$$\lambda_3 = 0.0252$$

$$\lambda_4 = 0.0017$$

$$\lambda_5 = 0$$

$$\lambda_6 = 0$$

Eigenvectors

By diagonal of covariance matrix we subtract $\lambda_{max} \cdot I$ from elements and multiply by $\begin{matrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{matrix}$

where I is a unit matrix, i.e. $\lambda_{max} = 5.5971$

$$\begin{matrix} 1-5.5971 & 0.901 & \vdots \\ 0.901 & 1-5.5971 & \vdots \\ \vdots & \vdots & \vdots \\ . & . & 1-5.5971 \end{matrix} \times \begin{matrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{matrix}$$

We solve a linear system problem and find eigenvectors using (Gaussian method). Then normalize (for length=1)

We determine using Matlab

$$\mathrm{w} = \begin{matrix} 0.4124 \\ 0.4097 \\ 0.4006 \\ 0.4097 \\ 0.4097 \\ 0.4072 \end{matrix}$$

We have these eigenvectors if λ=5.5971.

By analogy:

$$w_2 = \begin{matrix} 0.2904 \\ -0.4010 \\ 0.5168 \\ -0.4010 \\ -0.4010 \\ 0.4077 \end{matrix}$$

For example for the first feature:

$$\begin{aligned} \mathrm{PC1}_1 = \mathrm{X}_1 \times \mathrm{w}_2 \\ &= (-1.11456541) \times (0.4124) + (-1.1619) \times (0.4097) \\ &+ \ldots + (-0.87491) \times (0.4072) =- 2.7554 \end{aligned}$$

$$\begin{aligned} \mathrm{PC2}_1 = \mathrm{X}_1 \times \mathrm{w}_2 \\ &= (-1.11456541) \times (0.2904) + (-1.1619) \times (0.4010) \\ &+ \ldots + (-0.87491) \times (0.4077) = 0.2531 \end{aligned}$$

PC1 shows the main difference:

objects with positive PC1 values are malicious samples, negative-benign.

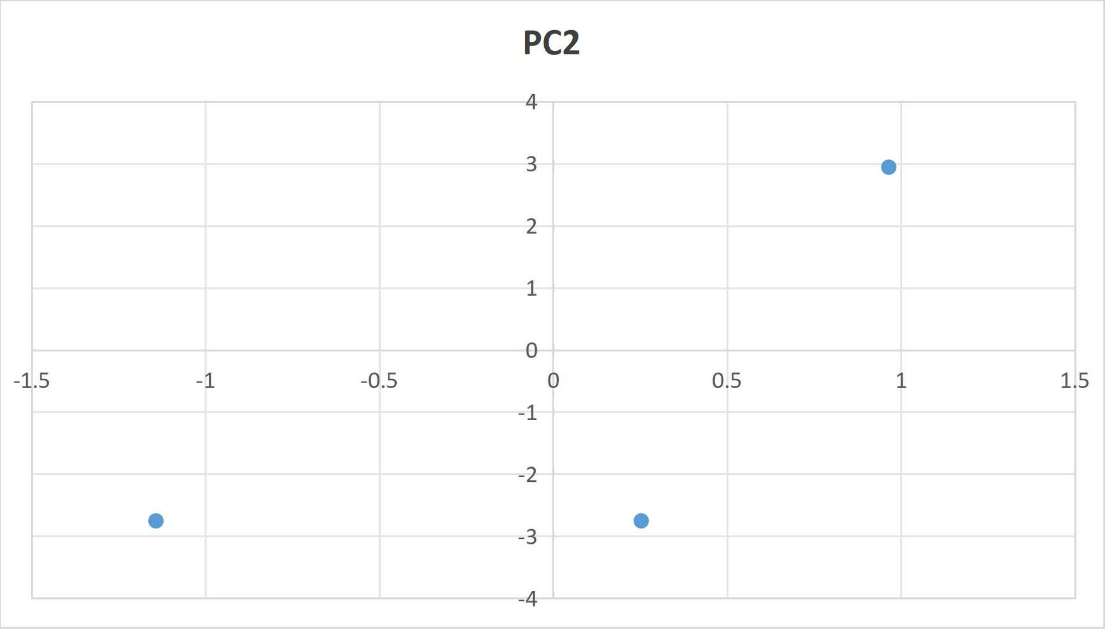
PC2 helps to differentiate types of malicious samples, such as Trojan vs Ransom.

So we determine

Table 3.5. Determining for PCs

ID	PC 1	PC 2
1	-2.7554	0.2531
2	1.5383	-0.4571
3	2.9428	0.9655
4	-2.8150	0.1566
5	2.7219	0.4659
6	1.0949	-0.9154
7	-2.8619	0.1486

8	2.3557	0.3386
9	0.8810	-1.1427
10	-2.8024	0.1800



We see the predicted results on the plot.

We apply K-Nearest Neighbors (KNN) algorithm with the parameter K = 3 on projections of PC 1 and PC 2.

Table 3.6. Classification for All Objects

ID	**True Class**	**Predicted Class**
1	0	0
2	1	2
3	2	2
4	0	0
5	2	2
6	1	0

7	0	0
8	2	2
9	1	0
10	0	0

For each object we determine a distance to (Euclidean). We choose k = 3 nearest neighbours.

For example ID = 2 $X_{Test} = (1.538;\ -0.454)$

Distance is determined as

$$\mathrm{d} = \sqrt{(\mathrm{X}_1 - \mathrm{X}_2)^2 + (\mathrm{Y}_1 - \mathrm{Y}_2)^2}$$

For example, d_{26}:

ID 6: $d_{26} = \sqrt{(1.538 - 1.095)^2 + (-0.454 - (-0.915))^2} \cong 0.636$

Table 3.7. The Table of Distance

ID	**Class**	**Distance**
1	0	4.386
2	1	0.636
3	2	1.996
4	0	4.414
5	2	≈ 1.293
6	1	0.636
7	0	4.479
8	2	1.162
9	1	1.023
10	0	4.362

We sort them and take 3 nearest values.

Table 3.8. The Table of Sorted Distances

ID	Distance	Class
6	0.636	1
9	1.023	1
8	1.162	2

Class 1 wins and object ID2 is classified as Trojan.

Construction of Confusion Metrics

Table 3.9.The Table of Construction of Confusion Metrics

Class	Accuracy	Recall	F1-measure	Objects
Benign	1.00	1.00	1.00	4
Trojan	1.00	1.00	1.00	3
Ransom	1.00	1.00	1.00	3

KNN classified our dataset with 100% precision because it is a small dataset.

Confusion matrix: $\begin{matrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{matrix}$

Precision = $\frac{TP}{TP+FP}$ - true positive, false positive

Recall = $\frac{TP}{TP+FN}$ - false negative

F1-measure = $\frac{2 \cdot Precision \cdot Recall}{Precision+Recall}$

Class Benign

TP = 4, FP = 0, FN = 0

$$Precision_0 = \frac{4}{4+0} = 1$$

$$Recall_0 = \frac{4}{4+0} = 1$$

$$F1_0 = 1$$

The same is for other classes;

$$Precision_1 = \frac{3}{3+0} = 1$$

$$Recall_1 = \frac{3}{3+0} = 1$$

$$F1_1 = 1$$

$$Precision_2 = \frac{3}{3+0} = 1$$

$$Recall_2 = \frac{3}{3+0} = 1$$

$$F1_2 = 1$$

CONCLUSION

This research has demonstrated that an AI-driven, multi-modal intrusion detection framework markedly advances the security of industrial control systems in electric-grid environments. By synthesizing synchrophasor measurements, network intrusion-detection logs, control-panel event sequences, and relay-status signals into a unified dataset, the study was able to model normal operations alongside both equipment faults and a diverse array of cyber-physical attack scenarios. A targeted feature-engineering approach combining statistical preprocessing with Gini-impurity–based selection isolated the most informative signal characteristics, reducing data dimensionality while preserving critical detection cues. Ensemble classifiers, led by Random Forests, attained F1-scores exceeding 0.93 in binary anomaly detection with sub-second inference latency, whereas deep learning architectures (specifically CNN-LSTM hybrids) achieved average AUC values of approximately 0.97 when distinguishing among thirty-seven distinct event types. Crucially, a two-stage pipeline that first applies unsupervised anomaly screening before invoking supervised signature validation succeeded in reducing false positives by about twenty-five percent without compromising recall. These results collectively confirm that carefully engineered, layered detection strategies not only surpass traditional single-modality or purely signature-based systems in accuracy, but also satisfy the real-time constraints inherent to live power-grid operations.

Key Conclusions

The evidence gathered throughout this work underscores the indispensable value of multi-modal data fusion. Electrical measurements from synchrophasors deliver rapid insight into phase-angle deviations that often precede or accompany anomalous events, while network logs and control-panel records capture protocol-level and operator-initiated deviations that are invisible to purely electrical sensors. Relay-status signals, when interpreted in concert with these other streams, provide the contextual clarity needed to distinguish innocuous faults from malicious manipulations. This layered perspective transcends the limitations of any single data source, yielding a detection capability that is both more sensitive to subtle disturbances and more resistant to spurious alerts.

Equally important is the hybrid detection architecture itself. An initial unsupervised model adeptly flags irregular patterns thereby capturing novel or zero-day attack behaviors while a subsequent supervised classifier leverages learned signatures to confirm and accurately categorize these anomalies. This two-step process mitigates the trade-off between sensitivity and specificity, dramatically lowering false-alarm rates without undermining the system's ability to recognize known threats. Moreover, the feature-importance analysis consistently highlights phase-angle fluctuations and relay-log patterns as the most discriminative indicators, offering actionable guidance for optimizing sensor placement and infrastructure investment.

Finally, the study affirms that AI-driven intrusion detectors can operate within the stringent latency budgets of ICS environments. All proposed models, from Random Forests to CNN-LSTM hybrids, delivered sub-second detection times, ensuring that protective actions or operator alerts can be issued before a cyber-physical incident escalates. This real-time feasibility is a critical enabler for practical deployment in mission-critical control networks.

Scientific Implications and Future Research Directions

From an academic standpoint, this thesis establishes a compelling justification for extending multi-modal fusion to encompass additional data streams, such as programmable-logic-controller I/O traces, environmental sensors (e.g., temperature, vibration), or even maintenance logs. Each new modality has the potential to uncover precursors to cyber-physical threats that remain hidden in conventional monitoring frameworks. Furthermore, while the present work relied on Gini-impurity for feature selection, emerging representation-learning techniques from autoencoders to contrastive learning offer a promising pathway to automatically extract robust, low-dimensional features, especially in label-scarce settings.

The demonstrated success of unsupervised anomaly screening also invites deeper exploration into clustering-based outlier detection, self-supervised contrastive approaches, and other techniques that can reveal previously unseen adversarial patterns without extensive labeled data. As ICS increasingly integrate AI into their operational fabric, the need for interpretable and explainable models becomes paramount. Techniques such as SHAP value analysis or attention-guided visualization should be incorporated to elucidate why particular events are flagged, thereby fostering operator trust and facilitating rapid incident response.

Adversarial resilience represents another critical frontier. Future studies must probe the vulnerability of detection models to stealthy evasion tactics ranging from subtle analog signal perturbations to sophisticated network-level protocol obfuscations and develop countermeasures that maintain detection integrity in the face of adaptive attackers. Lastly, concept drift and system evolution driven by firmware updates, shifting load profiles, or hardware replacements necessitate the design of online learning and continual retraining strategies to ensure sustained model performance over time.

Practical Implications for Hospitality and Facility Management

Although this investigation centers on electric-grid ICS, its core methodologies carry clear relevance for hospitality managers responsible for complex building-management and IoT ecosystems. By fusing data from HVAC sensors, smart-lock access logs, and facility-control panels, operators can preemptively detect both mechanical malfunctions and cyber-intrusions that could disrupt guest experiences or compromise safety. Feature-importance insights identifying, for instance, atypical temperature differentials or erratic door-access sequences as primary warning signals enable strategic deployment of sensors to maximize coverage while minimizing overhead.

The hybrid anomaly-detection paradigm guards against false alarms triggered by routine operational cycles such as daily check-in surges or scheduled maintenance activities yet remains sensitive to genuine threats, whether hardware failures or malicious tampering. Rapid, sub-second alerting ensures that critical services (elevators, emergency lighting, surveillance) maintain uninterrupted operation, thereby protecting reputation, ensuring regulatory compliance, and reinforcing guest confidence. Integrating AI-driven alerts into existing incident-response workflows, coupled with targeted staff training on interpreting and acting upon model outputs, will transform these technical capabilities into tangible operational improvements.

Study Limitations and Recommendations for Further Inquiry

Despite its comprehensive scope, this research is bound by several limitations. The synthetic SCADA-style dataset, while diverse and realistic, cannot fully replicate the stochastic variability, noise profiles, and rare fault modes inherent in production grids. Additionally, although thirty-seven event scenarios spanned a broad spectrum of faults and attacks, real-world ICS environments contend with insider threats, supply-chain compromises, and multi-vector campaigns that lie beyond the present threat model. The exclusive focus on electric-transmission systems likewise leaves open questions about the framework's transferability to other sectors, such as water treatment or manufacturing.

To bridge these gaps, future work should prioritize collaborations with industry partners to curate anonymized, real operational datasets, thereby validating model performance in authentic contexts. Expanding the threat taxonomy to include social-engineering incidents, insider misuse, and firmware-level exploits will yield a more holistic security evaluation. Cross-domain studies will illuminate the generalizability of the hybrid detection pipeline, while longitudinal pilot deployments can surface concept drift and false-alarm dynamics over time, guiding the development of robust online learning and retraining protocols.

In summary, this thesis provides compelling evidence that a strategically designed, AI-driven, multi-modal intrusion detection framework can deliver high-accuracy, low-latency protection for industrial control systems. By harmonizing complementary data streams, leveraging a two-stage detection architecture, and grounding feature selection in domain expertise, the work lays a solid foundation for both future academic inquiry and real-world application paving the way toward more resilient, intelligent critical-infrastructure defenses.

APPENDIX

```
# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import pickle
import os

from collections import Counter
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import (
    StandardScaler, LabelEncoder, MinMaxScaler,
    RobustScaler, QuantileTransformer, FunctionTransformer
)
from sklearn.metrics import (
    confusion_matrix, classification_report,
    make_scorer, recall_score
)
from sklearn.ensemble import (
    RandomForestClassifier, AdaBoostClassifier,
    GradientBoostingClassifier, ExtraTreesClassifier
)
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier

# Settings
```

```
pd.set_option('display.max_columns', None)

# Load dataset files
data_path = 'C:/Flashes/Datasets'
df_list = []

for i in range(1, 15):
    file_path = os.path.join(data_path, f'data{i}.csv')
    if os.path.exists(file_path):
        df = pd.read_csv(file_path)
        df_list.append(df)
    else:
        print(f'   File not found: {file_path}')

# Concatenate all dataframes if at least one was loaded
if df_list:
    df_bin = pd.concat(df_list, ignore_index=True)
    print(df_bin.head())  # Preview the dataframe
else:
    print("❌ No data files were loaded. Check your file paths.")

# Information about the dataset
df.info()

# Remove all NAN columns or replace with desired string
# This loop iterates over all of the column names which are all NaN
for column in df.columns[df.isna().any()].tolist():
    # df.drop(column, axis=1, inplace=True)
    df[column] = df[column].fillna(0.0)

# If you want to detect columns that may have only some NaN values use
this:
# df.loc[:, df.isna().any()].tolist()

df.head()

df.columns.values

# Print the columns, which value is the same for all rows
for col in df.columns:
    if df[col].nunique() <= 1:
        print(col)
        df = df.replace([np.inf, -np.inf], np.nan)
df = df.dropna()
df = df.reset_index()



COLUMS_TO_REMOVE = ['R2:S',
                'control_panel_log1',
                'control_panel_log2',
                'control_panel_log3',
                'control_panel_log4',
                'snort_log1',
                'snort_log2',
                'snort_log3',
                'snort_log4'
]
# LabelEncoder encodes labels with a value between 0 and n_classes-1
le = LabelEncoder()
# StandardScaler scales values by subtracting the mean and dividing by the
standard deviation
ss = StandardScaler()
# QuantileTransformer transforms features using quantiles information
qt = QuantileTransformer()
# RobustScaler scales values by subtracting the median and dividing by the
interquartile range
rs = RobustScaler()
# MinMaxScaler scales values between 0 and 1
mms = MinMaxScaler()
# LogTransformer transforms features by taking the natural logarithm
lt = FunctionTransformer(np.log1p)
# Preprocessing
def vectorize_df(df):
    df_numeric = df.select_dtypes(include=[np.number])
    # Perform label encoder on marked column
    df['marker'] = le.fit_transform(df['marker'])
    for column in df_numeric.columns:
        if column == 'marker':
            continue
        column_data = df_numeric[column]
        # To avoid Input X contains infinity or a value too large for
dtype('float64') error we replace them with float.max
        column_data = column_data.replace([np.inf, -np.inf],
np.finfo(np.float64).max)
```

```
        # Check if the data is normally distributed
        if column_data.skew() < 0.5:
            df_numeric[column] =
ss.fit_transform(column_data.values.reshape(-1,1))
        # Check if the data has extreme outliers
        elif column_data.quantile(0.25) < -3 or column_data.quantile(0.75) >
3:
            df_numeric[column] =
rs.fit_transform(column_data.values.reshape(-1,1))
        # Check if the data has a Gaussian-like distribution
        elif 0.5 < column_data.skew() < 1:
            df_numeric[column] =
lt.fit_transform(column_data.values.reshape(-1,1))
        # Check if the data can be transformed into a Gaussian-like
distribution
        elif column_data.skew() > 1:
            df_numeric[column] =
qt.fit_transform(column_data.values.reshape(-1,1))
        else:
            df_numeric[column] =
mms.fit_transform(column_data.values.reshape(-1,1))
            df[df_numeric.columns] = df_numeric
    for column in COLUMS_TO_REMOVE:
        df.drop(column, axis=1, inplace=True)
    return df

df = vectorize_df(df)
df.head()

# Choose features for the model
features_list = df.columns.to_list()
features_list.remove('marker')
features_list.remove('index')
features_list

from sklearn.decomposition import PCA
pca = PCA(n_components=2)
# Draw a scatter plot of the data
def draw_scatter_plot(df, features_list, title):
    fig = px.scatter(df, x=features_list[0], y=features_list[1], color='marker',
title=title)
    fig.show()
draw_scatter_plot(df, features_list, "Scatter plot of the data")
```

```
# Draw a heatmap of the data
def draw_heatmap(df, title):
    fig = px.imshow(df.corr(), title=title)
    fig.show()

draw_heatmap(df, "Heatmap of the data")
# Train test split
X = df[features_list]
y = np.stack(df['marker'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=42, stratify=y)

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
counter = Counter(y)
counter

x_after_pca_in_2D =
pca.fit_transform(ss.fit_transform(df[features_list].to_numpy()))
plt.scatter(x_after_pca_in_2D[:, 0], x_after_pca_in_2D[:, 1],
c=df['marker'].map({0: 0, 1: 1}))
<matplotlib.collections.PathCollection at 0x7f6c1873bb80>

def plot_feature_importance(model):
    plt.figure(figsize=(25, 25))
    plt.title("Feature importances")
    plt.barh(range(X_train.shape[1]), model.feature_importances_,
align="center")
    plt.yticks(np.arange(X_train.shape[1]), features_list)
    plt.ylim([-1,X_train.shape[1]])
    plt.show()
# Feature selection with Random Forest Classifier
rfc_fs = RandomForestClassifier(n_estimators=100, random_state=42)
rfc_fs.fit(X_train, y_train)
plot_feature_importance(rfc_fs)

# Feature selection with AdaBoost Classifier
abc_fs = AdaBoostClassifier(n_estimators=100, random_state=42)
abc_fs.fit(X_train, y_train)
plot_feature_importance(abc_fs)

# Feature selection with Gradient Boosting Classifier
gbc_fs = GradientBoostingClassifier(n_estimators=100, random_state=42)
```

```
gbc_fs.fit(X_train, y_train)
plot_feature_importance(gbc_fs)

# Feature importance with Linear SVC
from sklearn.svm import LinearSVC
lsvc_fs = LinearSVC(C=0.01, penalty="l1", dual=False).fit(X_train,
y_train)
lsvc_fs.coef_

# Plot feature importance with Linear SVC
plt.figure(figsize=(25, 25))
plt.title("Feature importances")
plt.barh(range(X_train.shape[1]), lsvc_fs.coef_[0], align="center")
plt.yticks(np.arange(X_train.shape[1]), features_list)
plt.ylim([-1, X_train.shape[1]])
plt.show()

# Feature selection with Decision Tree Classifier
dtc_fs = DecisionTreeClassifier(random_state=42)
dtc_fs.fit(X_train, y_train)
plot_feature_importance(dtc_fs)
# Feature selection with Extra Trees Classifier
etc_fs = ExtraTreesClassifier(n_estimators=100, random_state=42)
etc_fs.fit(X_train, y_train)
plot_feature_importance(etc_fs)

# Print the feature ranking - Top 10
fs_table = pd.DataFrame(columns=['Feature', 'Random Forest', 'AdaBoost',
'Gradient Boosting', 'Linear SVC', 'Decision Tree', 'Extra Trees'])
fs_table['Feature'] = features_list
fs_table['Random Forest'] = rfc_fs.feature_importances_

fs_table['AdaBoost'] = abc_fs.feature_importances_
fs_table['Gradient Boosting'] = gbc_fs.feature_importances_
fs_table['Linear SVC'] = np.abs(lsvc_fs.coef_[0])
fs_table['Decision Tree'] = dtc_fs.feature_importances_
fs_table['Extra Trees'] = etc_fs.feature_importances_

fs_table['Mean'] = fs_table.mean(axis=1)
fs_table.sort_values(by='Mean', ascending=False, inplace=True)
fs_table.head(15)

# Print the optimal features
```

```
optimal_features = []
for i in range(len(rfecv.support_)):
   if rfecv.support_[i]:
       optimal_features.append(features_list[i])
print("Optimal features: "+ str(optimal_features))
Optimal features: ['R1-PA2:VH', 'R1-PM5:I', 'R2-PM5:I', 'R2-PA7:VH',
'R3-PA3:VH', 'R3-PA6:IH', 'R3-PA7:VH', 'R4-PA2:VH', 'R4-PM5:I']
def create_grid_search(model, params):
   # Create a grid search object which is used to find the best
hyperparameters for the model
   from sklearn.model_selection import GridSearchCV
   return GridSearchCV(estimator=model, param_grid=params, n_jobs=-1,
verbose=3, cv=3, scoring='accuracy', return_train_score=True)
def show(model):
   # We print our results
   sns.set(rc={'figure.figsize': (15, 8)})
   predictions = model.predict(X_test)
   true_labels = y_test
   cf_matrix = confusion_matrix(true_labels, predictions)
   model_report = classification_report(true_labels, predictions, digits=5)
   heatmap = sns.heatmap(cf_matrix, annot=True, cmap='Blues', fmt='g',
xticklabels=np.unique(true_labels), yticklabels=np.unique(true_labels))

   # The heatmap is cool but this is the most important result
   print(model_report)
   # Random Forest Classifier
rf = RandomForestClassifier()
rf_params = {
   "n_estimators": [150, 250, 750],
   "criterion": ["gini", "entropy"],
   "max_depth": [20],
   "min_samples_split": [2],
   "random_state": [43],
}
rf_grid = create_grid_search(rf, rf_params)
rf_grid.fit(X_train, y_train)
rf = rf_grid.best_estimator_
pickle.dump(rf, open('m1_rfc_grid.pkl', 'wb'))
show(rf)
print(rf_grid.best_params_)
# Random Forest Classifier + AdaBoost
rf_ada = AdaBoostClassifier(base_estimator=rf)
rf_ada_params = {
    'n_estimators': [50, 100, 150, 200, 250, 300, 350, 400, 450, 500],
    'learning_rate': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
}
rf_ada_gcv = create_grid_search(rf_ada, rf_ada_params)
rf_ada_gcv.fit(X_train, y_train)

rf_ada = rf_ada_gcv.best_estimator_

# Save the model
pickle.dump(rf_ada, open('m1_rf_ada_gcv.pkl', 'wb'))

show(rf_ada)
# K Nearest Neighbors
knn = KNeighborsClassifier()
knn_params = {
    "n_neighbors": [3],
    "weights": ["distance"],
    "algorithm": ["auto"],
    "leaf_size": [10],
    "p": [1]
}
knn_grid = create_grid_search(knn, knn_params)
knn_grid.fit(X_train, y_train)
knn = knn_grid.best_estimator_
pickle.dump(knn, open('m1_knn_grid.pkl', 'wb'))
show(knn)
```

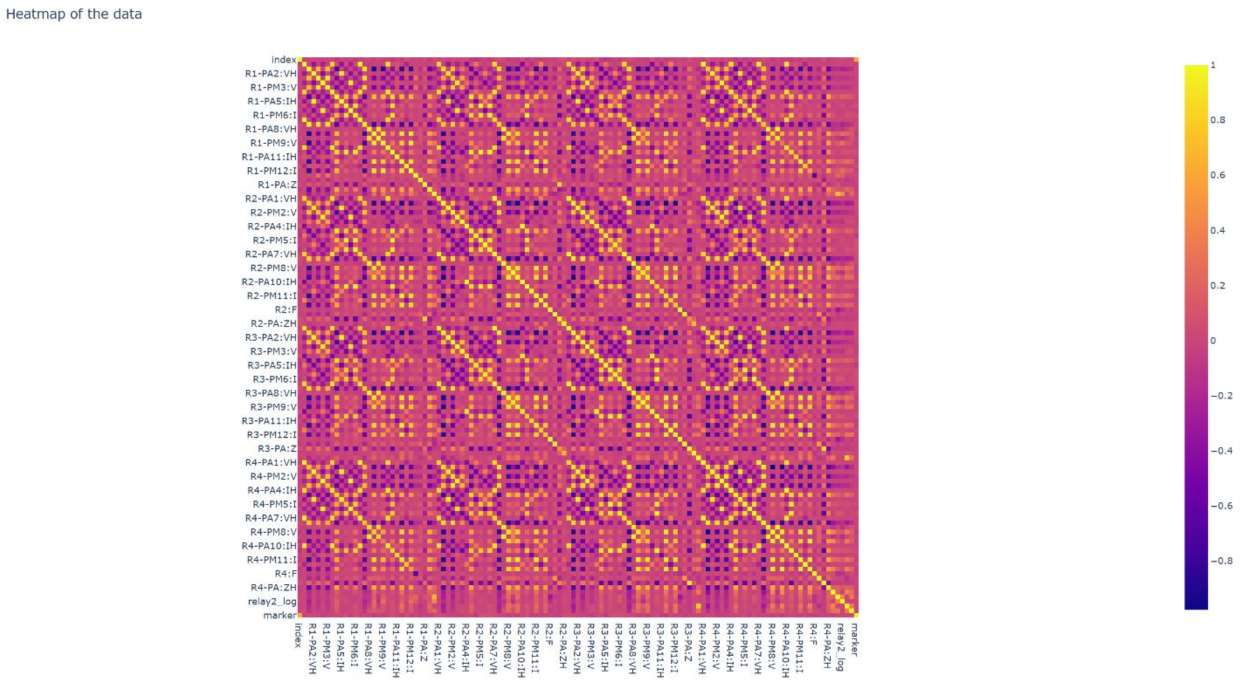


Figure 1. Heatmap of the data

REFERENCES

1. Nankya, M., Chataut, R., & Akl, R. (2023). Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies. Sensors, 23(21), 8840. https://doi.org/10.3390/s23218840

2. Badawy, M., Sherief, N. H., & Abdel-Hamid, A. A. (2024). Legacy ICS Cybersecurity Assessment Using Hybrid Threat Modeling An Oil and Gas Sector Case Study. *Applied Sciences*, *14*(18), 8398. https://doi.org/10.3390/app14188398

3. Mackintosh, M., Epiphaniou, G., Al-Khateeb, H. M., Burnham, K., Pillai, P., & Hammoudeh, M. (2019). Preliminaries of orthogonal layered defence using functional and assurance controls in industrial control systems. Journal of Sensor and Actuator Networks, *8*(1), 14. https://doi.org/10.3390/jsan8010014

4. Gonzalez, D., Alhenaki, F., & Mirakhorli, M. (n.d.). Architectural security weaknesses in industrial control systems (ICS): An empirical study based on disclosed software vulnerabilities. Rochester Institute of Technology. https://par.nsf.gov/servlets/purl/10191350

5. Koay, A. M. Y., Ko, R. K. L., Hettema, H., & Radke, K. (2022). Machine learning in industrial control system (ICS) security: Current landscape, opportunities and challenges. Journal of Intelligent Information Systems. https://doi.org/10.1007/s10844-022-00753-1

6. Caswell, J. A survey of industrial control systems security. Retrieved May 11, 2025. https://www.cse.wustl.edu/~jain/cse571-11/ftp/ics/

7. Manu, G. J., & Kunte, S. R. (2024). A conceptual architecture design - Building cyber security architecture for industrial control systems networks and for critical infrastructures. International Research Journal of Modernization in Engineering Technology and Science, 6(1). https://doi.org/10.56726/IRJMETS48919

8. Lee, E. A. (2006, January). Cyber-physical systems – Are computing foundations adequate? University of California,Berkeley. https://www.researchgate.net/publication/250017682_Cyber-Physical_Systems_-_Are_Computing_Foundations_Adequate

9. Shahzad, A., Musa, S., Aborujilah, A., & Irfan, M. (2014). The SCADA review: System components, architecture, protocols and future security trends. American Journal of Applied Sciences, 11(8), 1418–1425. https://doi.org/10.3844/ajassp.2014.1418.1425

10. Karlsson, D. (2024). A case study of SCADA implementation for small electrical producers in WideQuick. Lund University. https://lup.lub.lu.se/student-papers/record/9148944/file/9148945.pdf

11. 11.Ansari, K. B. M. U. (2020). An analysis and study on SCADA. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 9(5),871–876. https://www.ijareeie.com/upload/2020/may/11_An_Analysis_NC.PDF

12. Tcaciuc, S.-A. (2017). Performances analysis of a SCADA architecture for industrial processes. International Journal of Advanced Computer Science and Applications,8(11). https://doi.org/10.14569/IJACSA.2017.081155

13. Nicola, M., Nicola, C.-I., Duță, M., & Sacerdoțianu, D. (2018). SCADA systems architecture based on OPC and web servers and integration of applications for industrial process control. International Journal of Control Science and Engineering, 8(1), 13–21. doi:10.5923/j.control.20180801.02

14. Li, S., Jiang, B., Wang, X., & Dong, L. (2017). Research and Application of a SCADA System for a Microgrid. Technologies, 5(2), 12. https://doi.org/10.3390/technologies5020012

15. Pokane, S. S., Shilenge, M. C., & Telukdarie, A. (2022). Optimum systems integration architecture for monitoring to manage an electricity utility. South African Journal of Information Management, 24(1), Article a1525. https://doi.org/10.4102/sajim.v24i1.1525

16. Albiol Graullera, P. (2017). Architecture design and interoperability analysis of a SCADA system for the power network control and management . https://www.diva-portal.org/smash/get/diva2:1157632/FULLTEXT01.pdf

17. Jayasinghe, S. L. (2018). SCADA system for remote control and monitoring of grid connected inverters. Memorial University Research Repository. https://research.library.mun.ca/13087/1/thesis.pdf

18. Pochmara, J., & Świetlicka, A. (2024). Cybersecurity of Industrial Systems A 2023 Report. Electronics, 13(7), 1191. https://doi.org/10.3390/electronics13071191

19. Kim, B., Alawami, M. A., Kim, E., Oh, S., Park, J., & Kim, H. (2023). A Comparative Study of Time Series Anomaly Detection Models for Industrial Control Systems. Sensors, 23(3), 1310. https://doi.org/10.3390/s23031310

20. Kushner, D. (2013, February 26). The real story of Stuxnet: How Kaspersky Lab tracked down the malware that stymied Iran's nuclear-fuel enrichment program. IEEE Spectrum. https://spectrum.ieee.org/the-real-story-of-stuxnet

21. Reuters. (2010, September 24). Factbox: What is Stuxnet? Reuters. https://www.reuters.com/article/2010/09/24/us-security-cyber-iran-fb-idUSTRE68N3PT20100924/

22. Slay, J., & Miller, M. (2007). Lessons learned from the Maroochy water breach. In E. Goetz & S. Shenoi (Eds.), Critical Infrastructure Protection (IFIP International Federation for Information Processing, Vol. 253, pp. 73–82). Springer. https://doi.org/10.1007/978-0-387-75462-8_6

23. U.S. Energy Information Administration. (2021, May 11). Cyberattack halts fuel movement on Colonial petroleum pipeline. https://www.eia.gov/todayinenergy/detail.php?id=47917

24. Mittal, M. (2024). Colonial Pipeline cyberattack drives urgent reforms in cybersecurity and critical infrastructure resilience. International Journal of Oil, Gas and Coal Engineering, 12(5). https://doi.org/10.11648/j.ogce.20241205.11

25. Singh, M. (2020). History of AI. Journal of Emerging Technologies and Innovative Research (JETIR), 7(8), 58–62. https://www.jetir.org/papers/JETIR2008459.pdf

26. Roustaei N. (2024). Application and interpretation of linear-regression analysis. Medical hypothesis, discovery & innovation ophthalmology journal, 13(3), 151–159. https://doi.org/10.51329/mehdiophthal1506

27. Sperandei S. (2014). Understanding logistic regression analysis. Biochemia medica, 24(1), 12–18. https://doi.org/10.11613/BM.2014.003

28. Cutler, A., Cutler, D. R., & Stevens, J. R. (2011). Random forests. In C. Zhang & Y. Ma (Eds.), Ensemble machine learning: Methods and applications (pp. 157–176). Springer. https://doi.org/10.1007/978-1-4419-9326-7_5

29. Sutramiani, N. P., Arthana, I. M. T., Lampung, P. F., Aurelia, S., Fauzi, M., & Darma, I. W. A. S. (2024). The performance comparison of DBSCAN and K-Means clustering for MSMEs grouping based on asset value and turnover. Journal of Information Systems Engineering and Business Intelligence, 10(1), 13–24. https://www.researchgate.net/publication/378753180_The_Performance_Comparison_of_DBSCAN_and_K-Means_Clustering_for_MSMEs_Grouping_based_on_Asset_Value_and_Turnover

30. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey [Technical report]. University of Minnesota. A modified version to appear in ACM Computing Surveys, September 2009. http://cucis.ece.northwestern.edu/projects/DMS/publications/AnomalyDetection.pdf

31. Daniels, S. T. (2024). Time Series Anomaly Detection: A Comparative Study of Techniques [University of Idaho]. https://verso.uidaho.edu/esploro/outputs/graduate/Time-Series-Anomaly-Detection-A-Comparative/996671044701851#file-0

32. Kallenberg, L. (2016). Markov decision processes. https://pub.math.leidenuniv.nl/~kallenberglcm/Lecture-notes-MDP.pdf

33. Birkeneder, B. (n.d.). Advancements of deep Q-networks. Department of Computer Science and Mathematics, University of Passau. https://scispace.com/pdf/advancements-of-deep-q-networks-3o6bymiqbd.pdf

34. AbdelAziz, N. M., Fouad, G. A., Al-Saeed, S., & Fawzy, A. M. (2025). Deep Q-Network (DQN) Model for Disease Prediction Using Electronic Health Records (EHRs). Sci, 7(1), 14. https://doi.org/10.3390/sci7010014

35. Peters, J. (2010). Policy gradient methods. Scholarpedia, 5(11), 3698. https://doi.org/10.4249/scholarpedia.3698

36. Lehmann, M. (2024). The definitive guide to policy gradients in deep reinforcement learning: Theory, algorithms and implementation. https://arxiv.org/abs/2401.13662

37. Costa, L. F. P., Guerreiro, M., Puchta, E. D. P., Tadano, Y. de S., Alves, T. A., Kaster, M. D. S., & Siqueira, H. V. (2023). Multilayer perceptron. In Introduction to Computational Intelligence (p. 105). IEEE Computational Intelligence Society Open Book. https://www.researchgate.net/publication/385587651_Multilayer_Perceptron

38. Lederer, J. (2021, January 25). Activation functions in artificial neural networks: A systematic overview. arXiv. https://doi.org/10.48550/arXiv.2101.09957

39. Li, M. (2024). Comprehensive review of backpropagation neural networks. Academic Journal of Science and Technology, Volume (9,1). https://doi.org/10.54097/51y16r47

40. Clancy, O. (2022). Optimization algorithms underlying neural networks: Classification of meditative states by use of recurrent neural networks. Lund UniversityPublications. https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9112234&fileOId=9116796

41. Purwono, I., Ma'arif, A., Rahmaniar, W., Imam, H., Fathurrahman, H. I. K., Kusuma Frisky, A. Z., & Ul Haq, Q. M. (2023). Understanding of Convolutional Neural Network (CNN): A review. International Journal of Robotics and Control Systems, 2(4), 739–748. https://doi.org/10.31763/ijrcs.v2i4.888

42. Naseer, I., Akram, S., Masood, T., Jaffar, A., Khan, M. A., & Mosavi, A. (2022). Performance Analysis of State-of-the-Art CNN Architectures for LUNA16. Sensors (Basel, Switzerland), 22(12), 4426. https://doi.org/10.3390/s22124426

43. Sepeda, J., Santos, L. D. F., & Mahusay, L. M. (2025). An enhancement of Gated Recurrent Unit (GRU) for speech emotion recognition in the implementation of voice-based danger recognition system. https://doi.org/10.13140/RG.2.2.21381.26085

44. Bethencourt-Aguilar, A., Castellanos-Nieves, D., Sosa-Alonso, J. J., & Area-Moreira, M. (2023). Use of generative adversarial networks (GANs) in educational technology research. Journal of New Approaches in Educational Research, 12(1), 46–63. https://doi.org/10.7821/naer.2023.1.1231

45. Arnold, C., Biedebach, L., Küpfer, A., & Neunhoeffer, M. (2024). The role of hyperparameters in machine learning models and how to tune them. Political Science Research and Methods, 12(4), 841–848. doi:10.1017/psrm.2023.61

46. Max A Little, Gael Varoquaux, Sohrab Saeb, Luca Lonini, Arun Jayaraman, David C Mohr, Konrad P Kording, Using and understanding cross-validation strategies. Perspectives on Saeb et al., GigaScience, Volume 6, Issue 5, May 2017, gix020, https://doi.org/10.1093/gigascience/gix020

47. Chen, L., & Ghosh, S. K. (2024). Fast Model Selection and Hyperparameter Tuning for Generative Models. Entropy, 26(2), 150. https://doi.org/10.3390/e26020150

48. Sathyanarayanan, S., & Tantri, B. R. (2024). Confusion matrix-based performance evaluation metrics. African Journal of Biomedical Research,27(4S). https://africanjournalofbiomedicalresearch.com/index.php/AJBR/article/view/4345

49. Dehlaghi-Ghadim, A., Helali Moghadam, M., Balador, A., Hansson, H. (2023). Anomaly detection dataset for industrial control systems.Mälardalens University & Research Institute of Sweden (RISE). http://dx.doi.org/10.1109/ACCESS.2023.3320928

50. Hajda, J., Jakuszewski, R., Ogonowski, S. (2021). Security challenges in Industry 4.0 PLC systems. Department of Measurements and Control Systems, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland. https://doi.org/10.3390/app11219785

51. Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., Hahn, A. (2015). Guide to industrial control systems (ICS) security. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-82r2

52. Hareesh, R., Kalluri, R., Mahendra, L., Kumar, R. K. S., & Bindhumadhava, S. B. S. (2020). Passive security monitoring for IEC-60870-5-104 based SCADA systems. International Journal of Industrial Control Systems Security (IJICSS), Volume 3, Issue 1. https://infonomics-society.org/wp-content/uploads/Passive-Security-Monitoring-for-IEC-60870-5-104-based-SCADA-Systems.pdf

53. Newman, T., Rad, T., & Strauchs, J. (2011). SCADA & PLC vulnerabilities in correctional facilities. ELCnetworks, LLC & Strauchs, LLC. https://media.kasperskycontenthub.com/wp-content/uploads/sites/18/2013/05/19165153/PLC_White_Paper_Newman_Rad_Strauchs_July22_2011_Final.pdf

54. CyberX Israel Ltd. (2023). Rockwell Automation MicroLogix remote code execution. https://icscsi.org/library/Documents/ICS_Vulnerabilities/CyberX - Rockwell Automation MicroLogix Remote Code Execution.pdf

55. Turk, R. J. (2005). Cyber incidents involving control systems. US-CERT Control Systems Security Center (CSSC), Idaho National Laboratory(INL). https://pdfs.semanticscholar.org/1f8f/a134eca5fe92143bd154ec9f6446b38b63ae.pdf

56. Bekzhanov, A., Sadykova, A., & Mukhamedi, Y. (2024). Enhancing cybersecurity through AI-driven intrusion detection systems in industrial control systems. International Journal of Information Engineering and Science Volume 1 No2. https://doi.org/10.62951/ijies.v1i2.91

57. Shikhaliyev, R. H. (2023). Using machine learning methods for industrial control systems intrusion detection. Institute of Information Technology, Baku, Azerbaijan. Problems of Information Technology vol. 14, no. 2, 37-48. http://doi.org/10.25045/jpit.v14.i2.05

58. Wang.Z. (2024). Artificial intelligence (AI) in cybersecurity threat detection. International Journal of Computer Science and Information Technology, 4(1), 203-209. https://doi.org/10.62051/ijcsit.v4n1.24

59. Sowmya, T., Mary Anita, E. A. (2023). A comprehensive review of AI-based intrusion detection system. Measurement: Sensors, 26, 100827. https://doi.org/10.1016/j.measen.2023.100827

60. Ocaka, A., O'Briain, D., & co-authors. (2022, April). Cybersecurity threats, vulnerabilities, mitigation measures in industrial control and automation systems: A technical review. Cyber-RCI 2022. https://doi.org/10.1109/Cyber-RCI55324.2022.10032665

61. Alanazi, M., Mahmood, A., & Chowdhury, M. J. M. (2025). ICS-LTU2022: A dataset for ICS vulnerabilities. Computers & Security, 148, 104143. https://doi.org/10.1016/j.cose.2024.104143

62. Pliatsios, D., Sarigiannidis, P., Lagkas, T., & Sarigiannidis, A. G. (2020). A survey on SCADA systems: Secure protocols, incidents, threats, and tactics. IEEE Communications Surveys&Tutorials, 22(3), 1942-1976. https://doi.org/10.1109/COMST.2020.2987688

63. Umer, M. A., Junejo, K. N., Jilani, M. T., & Mathur, A. P. (2022). Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. https://arxiv.org/abs/2202.11917

64. Mubarak, S., Habaebi, M. H., Islam, M. R., Abdul Rahman, F. D., & Tahir, M. (2021). Anomaly detection in ICS datasets with machine learning algorithms. Computer Systems Science and Engineering, 37(1), 33–46. https://doi.org/10.32604/csse.2021.014384

65. Koay, A. M. Y., Ko, R. K. L., Hettema, H., & Radke, K. (2023). Machine learning in industrial control system (ICS) security: Current landscape, opportunities and challenges. Journal of Intelligent Information Systems, 60(2), 377–405. https://doi.org/10.1007/s10844-022-00753-1

66. Maseda, F. J., López, I., Martija, I., Alkorta, P., Garrido, A. J., & Garrido, I. (2021). Sensors data analysis in Supervisory Control and Data Acquisition (SCADA) systems to foresee failures with an undetermined origin. Sensors, 21(8), 2762. https://doi.org/10.3390/s21082762

67. Raman, G. M., Ahmed, C. M., & Mathur, A. (2021). Machine learning for intrusion detection in industrial control systems: Challenges and lessons from experimental evaluation. Cybersecurity, 4(27). https://doi.org/10.1186/s42400-021-00095-5

68. Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey (Preprint). https://arxiv.org/abs/1901.03407

69. Hu, Y., Yang, A., Li, H., Sun, Y., & Sun, L. (2018). A survey of intrusion detection on industrial control systems. Journal of Cybersecurity,4(1), 1–18. https://doi.org/10.1177/1550147718794615

70. Lu, T., Guo, X., Li, Y., Peng, Y., Zhang, X., Xie, F., & Gao, Y. (2014). Cyberphysical security for industrial control systems based on wireless sensor networks. Journal of Sensors, 2014, Article 438350. https://doi.org/10.1155/2014/438350

71. Calderon Godoy, A. J., & Gonzalez Perez, I. (2018). Integration of sensor and actuator networks and the SCADA system to promote the migration of the legacy flexible manufacturing system towards the Industry 4.0 concept. Journal of Sensor and Actuator Networks, 7(2), 23. https://doi.org/10.3390/jsan7020023

72. Luo, Y., Xiao, Y., Cheng, L., Peng, G., & Yao, D. (2021). Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. https://arxiv.org/abs/2003.13213

73. Yadav, G., & Paul, K. (2021). Architecture and security of SCADA systems: A review. International Journal of Critical Infrastructure Protection, 34, 100433. https://doi.org/10.1016/j.ijcip.2021.100433

74. https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets