Mitigating Cyber Threats in Smart Factories: A Hybrid Artificial Intelligence and Machine Learning Approach

# DEDICATION

# ACKNOWLEGMENT

# Table of content

[DEDICATION 2](#_Toc200520065)

[ACKNOWLEGMENT 3](#_Toc200520066)

[Table Of Content 4](#_Toc200520067)

[List Of Figures 8](#_Toc200520068)

[List Of Tables 9](#_Toc200520069)

[Abstract 10](#_Toc200520070)

[CHAPTER ONE 11](#_Toc200520071)

[Introduction 11](#_Toc200520072)

[1.1 Background Of The Study 11](#_Toc200520073)

[1.2 Problem Statement 13](#_Toc200520074)

[1.3 Aim And Objectives Of The Study 15](#_Toc200520075)

[1.3.1 Objectives 15](#_Toc200520076)

[1.4 Significance Of The Study 16](#_Toc200520077)

[1.5 Scope Of The Study 17](#_Toc200520078)

[1.6 Methodology For The Study 18](#_Toc200520079)

[Chapter Two 19](#_Toc200520080)

[Literature Review 19](#_Toc200520081)

[2.1 Introduction 19](#_Toc200520082)

[2.2 Conceptual Overview 19](#_Toc200520083)

[2.2.1 Smart Factories 19](#_Toc200520084)

[2.2.2 Cyber-Physical Systems (CPS) 21](#_Toc200520085)

[2.2.3 Cybersecurity In Industrial Environments 23](#_Toc200520086)

[2.2.4 Artificial Intelligence And Machine Learning In Cybersecurity 24](#_Toc200520087)

[2.2.5 Machine Learning 25](#_Toc200520088)

[2.2.6 RANDOM FOREST 31](#_Toc200520089)

[2.2.7 LONG SHORT-TERM MEMORY (LSTM) 31](#_Toc200520090)

[2.2.8 CONVOLUTIONAL NEURAL NETWORK 42](#_Toc200520091)

[2.2.9 Auto\_Encoder 46](#_Toc200520092)

[2.2.10 Hybrid AI Models 55](#_Toc200520093)

[2.3: Theoretical Framework 57](#_Toc200520094)

[2.3.1 Anomaly Detection Theory 57](#_Toc200520095)

[2.3.2 Defense-In-Depth Model 59](#_Toc200520096)

[2.4 Empirical Review 60](#_Toc200520097)

[2.4.1 AI And ML In Real-Time Threat Detection 60](#_Toc200520098)

[2.4.2 Performance Metrics From Recent Studies 62](#_Toc200520099)

[2.4.3 Practical Deployments In Industrial Settings 63](#_Toc200520100)

[2.5 Related Works 65](#_Toc200520101)

[2.6 Gap Identification 67](#_Toc200520102)

[CHAPTER THREE 69](#_Toc200520103)

[METHODOLOGY 69](#_Toc200520104)

[3.1 Introduction 69](#_Toc200520105)

[3.2 Research Design 69](#_Toc200520106)

[3.3 Data Collection And Dataset Description 70](#_Toc200520107)

[3.3.1 Dataset Prediction 70](#_Toc200520108)

[3.3.2 Justification For Dataset Selection 71](#_Toc200520109)

[3.4 Model Architecture 72](#_Toc200520110)

[3.4.1 Isolation Forest: Anomaly Detection Via Unsupervised Learning 72](#_Toc200520111)

[3.4.2 Autoencoder: Unsupervised Anomaly Detection 73](#_Toc200520112)

[3.4.3 LSTM + CNN: Hybrid Deep Learning For Temporal And Spatial Anomaly Detection 76](#_Toc200520113)

[3.4.4 Ensemble Integration Strategy: Isolation Forest, Autoencoder, And CNN-LSTM 78](#_Toc200520114)

[3.5 Model Training And Testing 79](#_Toc200520115)

[3.5.1 Data Preprocessing 79](#_Toc200520116)

[3.5.2 Training Strategy 80](#_Toc200520117)

[3.5.3 Model Evaluation 80](#_Toc200520118)

[3.6 Model Evaluation 81](#_Toc200520119)

[3.7 Ethical Considerations 82](#_Toc200520120)

[CHAPTER FOUR 83](#_Toc200520121)

[RESULT AND DISCUSSION 83](#_Toc200520122)

[4.1 Introduction 83](#_Toc200520123)

[4.2 Hardware And Software Configuration 83](#_Toc200520124)

[4.2.1 Hardware Configuration 83](#_Toc200520125)

[4.2.2 Software Configuration 84](#_Toc200520126)

[4.2.3 Implementation Environment 85](#_Toc200520127)

[4.3 Exploratory Data Analysis (EDA) And Preprocessing 85](#_Toc200520128)

[4.3.1 Dataset Overview And Structure 85](#_Toc200520129)

[4.3.2 Missing Values And Data Consistency 86](#_Toc200520130)

[4.3.3 Feature Distributions And Variability 86](#_Toc200520131)

[4.3.4 Visual Exploratory Data Analysis On The WADI Dataset 88](#_Toc200520132)

[4.3.5 Preprocessing Summary 91](#_Toc200520133)

[4.4 Model Training And Validation 92](#_Toc200520134)

[4.4.1 Isolation Forest 92](#_Toc200520135)

[4.4.2 Autoencoder 92](#_Toc200520136)

[4.4.3 LSTM-CNN Hybrid 93](#_Toc200520137)

[4.4.4 Ensemble Model: Weighted Fusion 93](#_Toc200520138)

[4.4.5 Validation Approach 94](#_Toc200520139)

[4.5 Model Evaluation Results 94](#_Toc200520140)

[4.5.1 Summary Of Results 95](#_Toc200520141)

[4.5.2 Isolation Forest Performance 95](#_Toc200520142)

[4.5.3 Autoencoder Performance 95](#_Toc200520143)

[4.5.4 CNN-LSTM Hybrid Performance 96](#_Toc200520144)

[4.5.5 Ensemble Performance 96](#_Toc200520145)

[4.5.6 Visual Interpretation And Diagnostic Tools 96](#_Toc200520146)

[4.6 Comparative Analysis Of Models 98](#_Toc200520147)

[4.6.1 Metric-Based Performance Comparison 98](#_Toc200520148)

[4.6.2 Visualization Comparison 98](#_Toc200520149)

[4.6.3 Critical Perspective 100](#_Toc200520150)

[4.7 Discussion Of Results 100](#_Toc200520151)

[4.7.1 Model Effectiveness And Complementarity 101](#_Toc200520152)

[4.7.2 Visualization Insights 101](#_Toc200520153)

[4.7.3 Real-World Implications For Smart Factories 102](#_Toc200520154)

[4.7.4 Comparative Perspective 102](#_Toc200520155)

[Chapter Five 104](#_Toc200520156)

[Summary, Conclusion, And Recommendation 104](#_Toc200520157)

[5.1 Summary 104](#_Toc200520158)

[5.2 Conclusion 105](#_Toc200520159)

[5.3 Recommendations 107](#_Toc200520160)

[5.4 Limitations And Suggestions For Future Work 108](#_Toc200520161)

[References 110](#_Toc200520162)

# List of figures

[Figure 2. 1 Different ML techniques and their required data (Mohammed et al., 2016) 28](#_Toc200530005)

[Figure 2. 2 Isolation Data Point 33](#_Toc200530006)

[Figure 2. 3 Random Selection of trees by Iso 33](#_Toc200530007)

[Figure 2. 4 Iso dividing to sub tree 34](#_Toc200530008)

[Figure 2. 5 Iso trying to isolate outliers 34](#_Toc200530009)

[Figure 2. 6 Iso creating data point an trees on different sides 35](#_Toc200530010)

[Figure 2. 7 Iso separated the outliers from the set 35](#_Toc200530011)

[Figure 2. 8 Iso moment split was occurring 36](#_Toc200530012)

[Figure 2. 9 separation still occurring in ISO using Box 36](#_Toc200530013)

[Figure 2. 10 calculation the split average 37](#_Toc200530014)

[Figure 2. 11 Iso finding the closest egde randomly 37](#_Toc200530015)

[Figure 2. 12 trying different order to ensure fairness in the random process 38](#_Toc200530016)

[Figure 2. 13 Reordering completed 38](#_Toc200530017)

[Figure 2. 14 The repeating module with four interacting layers (Colah, 2015) 39](#_Toc200530018)

[Figure 2. 15 LSTM cells and their operation 39](#_Toc200530019)

[Figure 2. 16 The Architecture of LSTM (Skymind, 2018a) 40](#_Toc200530020)

[Figure 2. 17 LSTM Pseudocode 42](#_Toc200530021)

[Figure 2. 18 Interaction among the elements of CNN 44](#_Toc200530022)

[Figure 2. 19 Operation of ReLU 45](#_Toc200530023)

[Figure 2. 20 After Pooling Layer, Flattened as Fully Connected Layer 46](#_Toc200530024)

[Figure 2. 21 Architecture of CNN Algorithm 46](#_Toc200530025)

[Figure 2. 22 Auto-encoders Architecture 47](#_Toc200530026)

[Figure 2. 23 Sparse autoencoder 50](#_Toc200530027)

[Figure 2. 24 Denoising Auto-encoders 53](#_Toc200530028)

[Figure 2. 25 Variational Auto encoders 54](#_Toc200530029)

[Figure 4. 1 Attack ratio in the dataset (class imbalance) 87](#_Toc200519915)

[Figure 4. 2 heatmap for Feature selection and correlation matrix from the WADI dataset 88](#_Toc200519916)

[Figure 4. 3 Time attack happened 89](#_Toc200519917)

[Figure 4. 4 checking 15 minutes attack density to confirm where there are higher spike 90](#_Toc200519918)

[Figure 4. 5 Average Sensor value during attack by Day 90](#_Toc200519919)

[Figure 4. 6 Z Scores of sensor anomalies during attack 91](#_Toc200519920)

[Figure 4. 7 heatmap of sensor attacks 92](#_Toc200519921)

[Figure 4. 8 Result of the Models 98](#_Toc200519922)

# List of tables

[Table 4. 1 Result summary of the models 96](#_Toc200519878)

# Abstract

The proliferation of cyber-physical systems in smart factories has significantly improved operational efficiency but has simultaneously exposed these environments to a growing array of sophisticated cyber threats. Traditional rule-based intrusion detection systems (IDS) often fall short in detecting novel or stealthy attacks. This study proposes a hybrid anomaly detection framework that combines three distinct machine learning models which are Isolation Forest, Autoencoder, and a CNN-LSTM architecture to improve threat detection in industrial control systems (ICS), with specific application to the WADI (Water Distribution) dataset obtained from the iTrust Lab. The framework was designed to exploit the complementary strengths of each model: Isolation Forest provides unsupervised outlier detection; Autoencoder detects deviations via reconstruction error; and CNN-LSTM captures both spatial and temporal patterns in time-series sensor data. Extensive exploratory data analysis (EDA) revealed multi-sensor correlation structures and time-bound attack windows, highlighting the complexity of anomaly behaviour in ICS. Model performance was evaluated using key metrics including AUC-ROC, Average Precision, Recall, Precision, and F1-score. Results demonstrated that: Isolation Forest achieved an AUC of 0.8886, recall of 79.9%, and precision of 60.2%; Autoencoder achieved AUC of 0.8085, with recall of 70.6% and precision of 54.7%; CNN-LSTM showed high precision of 97.4%, with recall of 63.5% and AUC of 0.6728; The ensemble model outperformed all, achieving an AUC of 0.8896, precision of 95.2%, and an F1-score of 70.4%. These results underscore the effectiveness of hybrid ensembles in reducing both false positives and missed detections. The proposed system demonstrates strong potential for real-time deployment in smart factory settings, offering a scalable and resilient cybersecurity solution aligned with Industry 4.0 objectives.

# CHAPTER ONE

# Introduction

## 1.1 Background of the Study

The onset of the Fourth Industrial Revolution, or Industry 4.0, has redefined manufacturing by integrating cyber-physical systems, Industrial Internet of Things (IIoT), artificial intelligence (AI), and machine learning (ML) into industrial operations (de Azambuja et al., 2023). At the heart of this transformation are smart factories, where interconnected sensors, actuators, control systems, and cloud-based services communicate seamlessly to optimize processes in real-time (Goyal, Rajawat, Solanki, Zaaba, and Long, 2023). However, this hyper-connectivity also exposes critical infrastructure to an evolving range of cyber threats, which can lead to operational shutdowns, data breaches, and physical damage.

Smart factories differ from traditional manufacturing plants due to their reliance on automated decision-making, real-time analytics, and edge computing. While these advancements promote agility and efficiency, they simultaneously increase the attack surface, making these environments more vulnerable to sophisticated cyberattacks such as Advanced Persistent Threats (APTs), ransomware, Distributed Denial of Service (DDoS) attacks, and zero-day exploits (Gajiwala, 2024). The convergence of operational technology (OT) and information technology (IT) within smart factories further compounds this risk, as traditional security frameworks were not designed for such complex, integrated systems (Zhang, Yang, and Yang, 2023).

Conventional cybersecurity solutions primarily rely on rule-based systems and signature detection, which are often reactive and ineffective against novel or polymorphic threats (Wang, Kuo, and Chen, 2021). These systems struggle to cope with the velocity, volume, and variety of data generated in real-time industrial environments (Jimmy, 2023). In contrast, AI and ML offer the potential for proactive threat detection, anomaly recognition, and autonomous response, making them vital tools in modern cybersecurity strategies.

AI-driven systems can monitor vast streams of network traffic and operational data to detect behavioral anomalies and potential attacks before they manifest. Machine learning algorithms, such as supervised learning (e.g., Random Forests, Support Vector Machines), unsupervised learning (e.g., clustering, autoencoders), and deep learning models (e.g., Long Short-Term Memory networks), have shown promising results in reducing false positives and improving real-time detection of cyber threats (Chen, Shen, Wang, Ke, and Xu, 2024).

A compelling case for the role of AI in cybersecurity is found in smart grid and IIoT environments, which share many vulnerabilities with smart factories. According to (Vilkelyte et al., 2024), AI and ML are increasingly employed in grid applications to detect and neutralize threats targeting interconnected components. The paper emphasizes the need for intelligent and adaptive cybersecurity architectures capable of evolving with the threat landscape.

In smart factory settings, AI enhances incident response times, automates intrusion detection systems (IDS), and minimizes human error in threat analysis. For example, (Elbes, Hendawi, Alzu’Bi, Kanan, and Mughaid, 2023) showed that machine learning models could outperform traditional methods in threat detection across multiple attack vectors, including phishing, malware, and insider threats.

Despite its advantages, implementing AI/ML in industrial environments is not without challenges. Key issues include the need for high-quality, labeled datasets, computational overhead, explainability of model decisions, and defense against adversarial AI attacks (Asere, Nuga, and Medugu, 2025). Additionally, legacy systems in older factories may not support the computational requirements of modern AI-based security solutions, making integration and scalability difficult.

To overcome these obstacles, researchers have begun exploring hybrid AI models that combine multiple learning paradigms. These hybrid systems typically integrate supervised, unsupervised, and deep learning methods to improve detection accuracy and reduce false positives. For instance, combining Random Forest classifiers for known threats, autoencoders for anomaly detection, and LSTMs for temporal pattern recognition offers a multi-dimensional defense mechanism tailored for complex smart factory networks (Diaba, 2024).

Another pressing issue is data localization and privacy. Smart factories operating in different regulatory jurisdictions must ensure compliance with data protection laws such as the GDPR. AI-driven security systems must be designed to uphold these standards while processing massive amounts of sensitive production and operational data (Gajiwala, 2024).

As digitalization accelerates, the economic and safety risks of cybersecurity breaches also grow. The use of AI to automate detection and response is not just a technical enhancement; it is increasingly a business imperative. According to Muheidat, Mallouh, Al-Saleh, Al-Khasawneh, and Tawalbeh, 2024), AI improves cost-efficiency by reducing manual analysis and downtime and offers real-time decision-making capabilities that legacy systems cannot match.

In response to these challenges and opportunities, this study aims to develop a hybrid AI/ML model tailored to smart factory cybersecurity. The proposed model will be trained and evaluated using publicly available industrial datasets, applying ensemble methods for optimized threat detection. This approach addresses critical gaps in current research by emphasizing real-world applicability, low-latency response, and compatibility with industrial control systems.

In conclusion, smart factories represent both the pinnacle of industrial automation and a growing frontier for cyber threats. Traditional security mechanisms are insufficient in this new paradigm. Artificial intelligence and machine learning provide a robust framework for proactive, scalable, and adaptive cybersecurity. However, to realize their full potential, these technologies must be strategically integrated through hybrid architectures that are context-aware, resource-efficient, and ethically sound.

## 1.2 Problem Statement

The integration of Industry 4.0 technologies into manufacturing, particularly the convergence of cyber-physical systems (CPS), Industrial Internet of Things (IIoT), and smart automation, has significantly increased both operational efficiency and the cybersecurity attack surface in smart factories. As factories become increasingly interconnected, they are also more vulnerable to sophisticated and evolving cyber threats, including Advanced Persistent Threats (APTs), ransomware, zero-day vulnerabilities, and insider threats (Jimmy, 2023). These attacks not only compromise data confidentiality and operational integrity but can lead to physical disruptions, safety hazards, and significant financial losses.

Traditional cybersecurity approaches, largely rule-based, signature-driven, and manually configured, struggle to detect and respond to these threats in real-time. They are inherently reactive and incapable of managing the dynamic, high-velocity, and heterogeneous data environments typical of smart factory ecosystems (Gajiwala, 2024). Furthermore, these legacy systems generate high false positive rates, which contribute to alert fatigue among security personnel and delayed threat response, both of which can have catastrophic consequences in real-time production environments (Elbes et al., 2023).

While Artificial Intelligence (AI) and Machine Learning (ML) have shown immense potential in enhancing cybersecurity by enabling predictive analytics, anomaly detection, and automated incident response, current implementations remain fragmented and limited. Standalone AI models often lack the robustness, contextual adaptability, or real-time responsiveness needed for industrial control systems. In many instances, these models are either trained on irrelevant datasets, fail to adapt to evolving attack vectors, or cannot operate efficiently within the constraints of legacy manufacturing infrastructure (Diaba, 2024).

Moreover, cyber threats are becoming increasingly multi-dimensional, requiring detection mechanisms that can identify both known (signature-based) and unknown (behavioral or zero-day) attacks across various temporal and spatial layers of factory networks (Chen et al., 2024). Single-method AI approaches (e.g., only using Random Forest or LSTM) fall short of this requirement due to their inherent limitations in coverage and adaptability.

Consequently, there is a critical need for a hybrid AI/ML model that can integrate the strengths of multiple learning paradigms, supervised for known threats, unsupervised for anomalies, and deep learning for sequential patterns, to deliver comprehensive, real-time cybersecurity protection in smart factories (Vilkelytė et al., 2024).

Therefore, the core problem this research seeks to address is the absence of a scalable, context-aware, and empirically validated hybrid AI and machine learning approach specifically optimized for mitigating diverse cyber threats in smart factory environments. Without such targeted innovations, smart factories will continue to remain highly vulnerable, jeopardizing the reliability, safety, and competitive edge promised by Industry 4.0.

## 1.3 Aim and Objectives of the Study

The aim of this study is to develop and evaluate a hybrid Artificial Intelligence (AI) and Machine Learning (ML) model designed to effectively detect, prevent, and mitigate cyber threats in smart factory environments.

### 1.3.1 Objectives

1. To critically analyze existing applications of AI and ML in smart factory cybersecurity, evaluating their strengths, weaknesses, and limitations across different industrial control system contexts.
2. To design and develop a hybrid AI/ML-based cybersecurity framework, combining supervised learning (e.g., Random Forest), unsupervised learning (e.g., Autoencoders), and deep learning (e.g., Long Short-Term Memory networks) to handle both known and unknown threats across time-series factory data.
3. To implement the hybrid model using real-world smart factory datasets, such as network logs, sensor outputs, and attack simulations from Water Distribution Dataset acquired from Itrust Centre for Research in Cybersecurity.
4. To evaluate and benchmark the hybrid model’s performance against conventional or standalone AI models using predefined metrics including accuracy, precision, recall, F1-score, false positive rate, and detection time.

## 1.4 Significance of the Study

The significance of this study lies in its direct response to the growing inadequacy of traditional cybersecurity approaches in protecting smart factory environments. As smart factories adopt interconnected technologies such as industrial IoT, edge computing, and cyber-physical systems, their vulnerability to cyberattacks increases dramatically. Conventional rule-based systems are no longer capable of addressing the velocity and complexity of cyber threats targeting these real-time, data-intensive environments (Jimmy, 2023). This research is significant as it proposes a novel hybrid AI/ML model that offers adaptive, scalable, and intelligent threat detection tailored specifically to the unique architecture and needs of smart factories.

Furthermore, this study addresses the critical performance limitations observed in current standalone AI models, which often fail to generalize across industrial scenarios due to poor adaptability or lack of temporal analysis capabilities. By integrating supervised, unsupervised, and deep learning techniques, the proposed model aims to overcome these limitations and provide more robust detection of both known and novel threats (Chen et al., 2024). This makes the study particularly relevant for industries relying on continuous production and real-time data integrity, where even minor delays or false positives can lead to operational losses or safety hazards.

The research also contributes to bridging the implementation gap by focusing on deployment feasibility in real-world industrial contexts. Issues such as integration with legacy systems, computational efficiency, and adversarial robustness are explored, making the study applicable to both high-tech and resource-constrained factory settings (Asere et al., 2025). In doing so, it supports the development of cybersecurity infrastructures that are not only technically advanced but also practical and sustainable in operational environments.

In a broader context, the study’s outcomes are valuable to cybersecurity professionals, industrial engineers, and policy-makers aiming to strengthen digital resilience in manufacturing. The empirical evidence generated through performance evaluation provides a foundation for more informed decisions on deploying AI-based security frameworks in critical infrastructure. Moreover, by contributing to the growing academic discourse on hybrid AI security systems, the research lays groundwork for future innovations in autonomous, intelligent cybersecurity models (Vilkelytė et al., 2024).

## 1.5 Scope of the Study

This study is specifically scoped to the technical evaluation and implementation of a hybrid Artificial Intelligence and Machine Learning (AI/ML) model for detecting and mitigating cyber threats in smart factory environments. The research focuses on industrial settings characterized by interconnected cyber-physical systems (CPS), IoT networks, and real-time data processing architectures, which collectively form the foundation of Industry 4.0. It emphasizes the development, simulation, and performance benchmarking of a multi-layered AI model integrating supervised (e.g., Random Forest), unsupervised (e.g., Autoencoder), and deep learning (e.g., LSTM) techniques.

The scope is limited to technical threat detection mechanisms, excluding broader organizational factors such as cybersecurity policy, employee behavior, regulatory compliance, and economic cost analysis. Additionally, while AI and ML systems are explored, the ethical and legal implications of AI use in cybersecurity, such as bias in model predictions or data governance, are only discussed to the extent they affect technical deployment and model reliability (Asere et al., 2025).

The model will be trained and tested using a water distribution dataset, from <https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/>, an iTrust research for Cybersecurity, which simulate attack scenarios relevant to industrial control systems. As such, the study does involve direct data collection from live industrial networks collected in 2019.

Lastly, the proposed solution is scoped for simulation-based validation rather than field deployment. It is intended as a prototype for future deployment and does not extend to hardware implementation on factory equipment or edge devices. Nonetheless, scalability and resource constraints typical of industrial edge environments are considered in model selection and architecture design to enhance future applicability (Vilkelytė et al., 2024).

## 1.6 Methodology for the Study

This study adopts a quantitative, simulation-based research methodology aimed at designing, developing, and evaluating a hybrid Artificial Intelligence (AI) and Machine Learning (ML) model for cyber threat detection in smart factory environments. The choice of methodology is rooted in the need for empirical validation of model performance across multiple threat dimensions, accuracy, latency, and robustness, without disrupting real-world industrial operations.

The experimental framework involves offline analysis using the WADI (Water Distribution) dataset, sourced from the iTrust Laboratory at the Singapore University of Technology and Design (SUTD). The WADI dataset is a real-world industrial dataset derived from a scaled-down water treatment testbed that mimics industrial control system (ICS) operations, including programmable logic controllers (PLCs), SCADA systems, and networked sensors. The dataset includes both normal operational data and labeled cyberattacks, making it suitable for supervised, unsupervised, and temporal pattern learning. Access to this dataset was obtained through formal request and approval from iTrust, ensuring ethical and authorized data usage.

Data preprocessing includes cleaning, normalization, and feature selection. The model will be trained using 70% of the data and validated on 30%, with performance evaluated using metrics such as accuracy, precision, recall, F1-score, false positive rate, and detection time (Chen et al., 2024). Model implementation will be done in Python using Scikit-learn, TensorFlow/Keras, and evaluated in Jupyter Notebook. Results will be compared to standalone models to validate the benefit of hybridization.

# Chapter two

# Literature review

## 2.1 Introduction

The increasing digitalization of manufacturing processes under Industry 4.0 has elevated the importance of cybersecurity in smart factories. These environments rely heavily on interconnected systems, cyber-physical systems (CPS), industrial IoT (IIoT), and cloud-based infrastructures, which, while enhancing efficiency, significantly expand the potential attack surface for cyber threats. The convergence of operational technology (OT) and information technology (IT) introduces unique vulnerabilities that traditional security mechanisms struggle to address.

Traditional rule-based and signature-based detection systems are often reactive and unable to identify zero-day threats or subtle anomalies in real-time, dynamic factory environments (Gajiwala, 2024). In response to these limitations, researchers and practitioners are increasingly turning to Artificial Intelligence (AI) and Machine Learning (ML) to provide proactive, adaptive, and scalable cybersecurity solutions.

This chapter reviews existing literature across five critical areas: the conceptual foundation of smart factory security, theoretical models supporting AI-driven detection, empirical applications of ML techniques, analysis of related works, and identified gaps in current research. The objective is to establish a robust knowledge base that justifies the development of a hybrid AI/ML model tailored specifically to cyber threat detection in smart factories.

## 2.2 Conceptual Overview

### 2.2.1 Smart Factories

Smart factories represent the core technological innovation driving Industry 4.0, an era characterized by the fusion of cyber-physical systems (CPS), the Industrial Internet of Things (IIoT), big data analytics, and artificial intelligence (AI) into manufacturing processes. These factories function as highly digitized production environments where machines, sensors, and control systems are interconnected to enable real-time data sharing, autonomous decision-making, and self-optimization of operations (Mosleuzzaman, Arif, and Siddiki, 2024).

Unlike traditional factories, smart factories operate on the foundation of integrated information and operational technologies. CPS facilitates the connection between the physical and digital world, allowing physical assets to be monitored and controlled through computational algorithms. When integrated with IoT, these systems become capable of collecting large volumes of data from sensors and actuators across the production floor, which can then be processed in real time to optimize decision-making (Exploring the Foundation of Smart Factories in Industry 4.0: A Conceptual Review, 2024)

The functional advantages of smart factories are well established. These include improved operational efficiency, reduced downtime, predictive maintenance, real-time quality control, and increased production flexibility. For example, embedded AI systems allow for adaptive learning based on historical production patterns, enabling proactive fault detection and response without human intervention (Musaeva, Vyachina, and Aliyeva, 2024). Moreover, smart factories enable mass customization through flexible production lines that adjust based on user demand and product specifications.

However, despite these benefits, the transition to smart manufacturing introduces complex challenges, particularly in cybersecurity. Smart factories are inherently exposed to cyber risks due to their dependence on interconnected digital infrastructure. Every sensor, actuator, and communication protocol introduce a potential entry point for malicious actors. The reliance on real-time data also means that any disruption can have immediate and severe consequences on production, safety, and supply chain continuity (Masum, 2023).

Moreover, the fusion of IT (information technology) and OT (operational technology) in smart factories complicates the security landscape. Unlike IT systems, OT components are often designed with long operational lifespans and minimal consideration for cybersecurity, making them vulnerable to attacks such as ransomware, DDoS, and advanced persistent threats (APTs) (Yi and Jeong, 2022). Furthermore, OT networks typically lack built-in encryption or access control mechanisms, and updates may be limited due to system stability requirements.

The need for secure, adaptive, and intelligent cybersecurity frameworks has become urgent. As highlighted by (Jamai, Ben Azzouz, and Saidane, 2020), IIoT devices in smart factories are particularly susceptible to attacks because of their limited computing power, outdated firmware, and frequent default configurations. Given these vulnerabilities, smart factories have become high-value targets for cybercriminals and even nation-state actors seeking to exploit industrial control systems for espionage or sabotage.

In summary, smart factories are transformative in their ability to create intelligent, agile, and efficient production environments. However, this transformation comes at the cost of increased cybersecurity complexity. Addressing these vulnerabilities demands the deployment of advanced threat detection mechanisms, particularly those powered by AI and ML, capable of monitoring large-scale data flows, identifying anomalies, and responding autonomously to evolving threats in real time.

### 2.2.2 Cyber-Physical Systems (CPS)

Cyber-Physical Systems (CPS) are foundational to smart factory operations, serving as the integrative bridge between the digital and physical realms of manufacturing. These systems comprise tightly coupled components of computation, networking, and physical processes that interact in real time to monitor, analyze, and control industrial activities. In a smart factory context, CPS enables the automation, flexibility, and adaptability necessary for high-efficiency and data-driven production environments (Yao et al., 2019).

CPS is not just an enabling technology; it is the structural backbone of Industry 4.0. Its architecture typically integrates industrial IoT (IIoT), real-time sensors, edge computing, and cloud services to create an interconnected ecosystem where every physical device is mirrored by a virtual counterpart. These digital twins communicate continuously, allowing for decentralized control and automated decision-making. As noted by Renteria-Marquez, Basaldua, Aguirre, Lara-Medrano, and Tseng (2024), the use of CPS in digital twins enables predictive scheduling, remote diagnostics, and real-time control of production systems.

However, the integration of CPS into manufacturing also introduces significant cybersecurity challenges. Unlike traditional IT systems, CPS must maintain both digital security and physical system integrity. Any cyberattack on a CPS can potentially cause physical damage, production loss, or even threaten human safety. These concerns are amplified in industrial environments where uptime and deterministic response are critical. The complexity and heterogeneity of CPS components, ranging from PLCs and SCADA systems to cloud-based analytics, further increase the attack surface and make end-to-end security more difficult to enforce (Simonthomas, Subramanian, and Mathiew, 2024).

As pointed out by Contreras et al. (2023), CPS in industrial environments face a wide range of cyber threats, including denial-of-service attacks, spoofing, and ransomware. The decentralized and distributed nature of CPS makes traditional perimeter-based security models ineffective. Therefore, there is a pressing need for embedded, context-aware, and autonomous security mechanisms that operate within the CPS architecture itself.

Moreover, CPS presents a unique trade-off between performance and security. Ensuring low-latency, high-reliability communication is essential in real-time manufacturing, but adding encryption, authentication, and intrusion detection can introduce delays and overhead. As Jain (2021) emphasizes, one of the core challenges in CPS security is maintaining this balance without degrading system performance or responsiveness.

To address these risks, architectural improvements to CPS are being proposed. For instance, new frameworks like the 8C architecture add dimensions such as coalition, customer integration, and content awareness to enhance system resilience and interoperability within smart factory environments (Jiang, 2018). These enhancements allow systems to become more adaptive to dynamic operational and cybersecurity conditions.

In conclusion, CPS is both a powerful enabler and a critical vulnerability point in smart factories. While it underpins the operational intelligence of modern manufacturing, it also introduces complex cybersecurity threats that demand advanced, multi-layered protection strategies. Understanding and securing CPS is therefore essential not just for factory productivity but also for maintaining industrial safety and continuity in the face of growing cyber risks.

### 2.2.3 Cybersecurity in Industrial Environments

As industrial environments adopt advanced digital technologies under Industry 4.0, cybersecurity has emerged as one of the most critical and complex challenges. Smart factories now rely on deeply interconnected systems including cyber-physical systems (CPS), industrial Internet of Things (IIoT), cloud computing, and machine learning tools. While these technologies increase efficiency, automation, and flexibility, they also significantly expand the attack surface, exposing critical infrastructure to a wide range of cyber threats (Masum, 2023).

Unlike traditional IT systems, industrial environments operate on both digital and physical layers, making cyberattacks potentially destructive in real-time operations. Attacks targeting industrial control systems (ICS), programmable logic controllers (PLCs), or SCADA components can directly disrupt production, damage physical equipment, or endanger human safety. These risks are intensified by the inherent vulnerabilities of OT (Operational Technology), such as legacy software, weak access controls, and limited cryptographic protection (Juarez, 2019).

One of the critical gaps in securing industrial environments is the disconnect between IT and OT security priorities. IT focuses on data confidentiality and integrity, while OT emphasizes real-time availability and process continuity. This divergence makes unified cybersecurity strategies difficult to implement. Moreover, updating or patching OT systems is often delayed or avoided entirely due to fears of operational disruption (Corallo, Lazoi, Lezzi, and Luperto, 2022).

A growing body of research emphasizes the need for holistic, real-time cybersecurity frameworks that incorporate anomaly detection, predictive analytics, and automated response mechanisms. These systems must be integrated into the production environment without disrupting operations, which demands lightweight, adaptive solutions. For example, AI-driven security models can help identify abnormal behavior patterns and zero-day threats before they escalate (Goyal et al., 2023).

However, deploying AI and machine learning in industrial cybersecurity is not without its challenges. High-quality labeled datasets for training are often scarce, and real-time processing demands optimized architectures that balance performance and computational load. Additionally, adversarial attacks that manipulate AI predictions pose new risks, especially when applied to critical infrastructure control systems (da Silva Oliveira and Santos, 2022).

Regulatory frameworks like ISA/IEC 62443 are increasingly promoted to standardize industrial cybersecurity practices, offering layered defense models and certification guidelines. Still, many organizations fall short in implementation due to cost, complexity, or lack of skilled personnel. As Kulinich (2023) notes, the management of cybersecurity in smart factories must also evolve, integrating strategic planning, workforce training, and continuous threat assessment into operational protocols.

In conclusion, cybersecurity in industrial environments is a multi-faceted challenge that requires more than just technological upgrades. It demands integrated strategies combining AI-powered threat detection, cross-domain policy alignment, resilient system architectures, and continuous workforce education. Without these, the very technologies designed to enhance productivity could become the entry points for catastrophic industrial failures.

### 2.2.4 Artificial Intelligence and Machine Learning in Cybersecurity

Artificial Intelligence (AI) and Machine Learning (ML) have become transformative tools in cybersecurity, particularly in the context of smart factories where cyber-physical systems and IIoT generate vast streams of real-time data. These technologies allow for predictive analysis, anomaly detection, and autonomous decision-making in environments where manual intervention is too slow or unreliable for effective threat mitigation (Hartmann, Brock, Kühn, and Dumitrescu, 2024).

In contrast to traditional rule-based systems, which require known threat signatures, ML models learn patterns from data, enabling them to detect unknown or evolving threats. Supervised learning algorithms such as Random Forest and Support Vector Machines are often used to classify network traffic or detect intrusions, while unsupervised approaches like clustering and autoencoders identify previously unseen anomalies in complex industrial datasets (Diaba, 2024). Deep learning models, particularly LSTM networks and convolutional autoencoders, further enhance performance by capturing temporal and spatial patterns in factory network traffic (Milić, 2024).

AI-enabled cybersecurity systems in smart factories also contribute to reducing false positives and automating threat responses. According to Siam, Hassan, and Bhuiyan (2025), combining ML with anomaly detection and natural language processing can help identify subtle signs of phishing, insider threats, and malware activity, tasks where traditional systems often fail or overload operators with alerts.

Despite these benefits, challenges remain in real-world deployments. As observed by Gavrovska and Samcovic (2020), issues such as insufficient labeled data, high computational cost, and integration with legacy factory systems continue to hinder wide-scale adoption. Furthermore, AI models themselves can be targets of adversarial attacks, where inputs are subtly manipulated to deceive classification systems.

Real-world implementation also involves socio-technical barriers. A study of over 50 smart factory deployments highlighted that beyond technological readiness, factors like lack of management support, data governance concerns, and workforce resistance can undermine AI initiatives (Hartmann et al., 2024). This emphasizes the need for a systems-level approach that includes technical innovation, human training, and organizational readiness.

In conclusion, AI and ML offer powerful solutions for securing smart factory environments, particularly through real-time monitoring, predictive threat detection, and autonomous response. However, to fully realize their potential, future research must focus on improving model transparency, adversarial robustness, and deployment strategies that are sensitive to operational and organizational constraints.

### 2.2.5 Machine Learning

Arthur Samuel first used the term "machine learning" in 1959. (Samuel, 1959). "A computer programme is said to learn from experience E with regard to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, increases with experience E," said Tom M. Mitchell in a more formal definition that is often used (Mitchell, 1999). Instead of describing the discipline in terms of cognition, this characterization of the tasks with which machine learning is concerned gives a fundamentally operational definition.

Machine learning in artificial intelligence is the investigation and development of algorithms that can learn from and make predictions on data. These algorithms circumvent purely static programme instructions by generating predictions or judgments based on the available data. Computational statistics, which similarly focuses on generating predictions using computers, is closely connected to machine learning (Rodrigues et al., 2018). It has optimization that provides the discipline with methodologies, theories, and application fields. Data mining and machine learning are frequently confused, with data mining focusing more on exploratory data analysis, or unsupervised learning. Unsupervised machine learning is another option that may be used to discover significant abnormalities after learning and establishing baseline behavioural profiles for different entities. Machine learning is a technique used in data analytics to create intricate models and algorithms that are conducive to predictions (Doherty et al., 2016). The use of human expert knowledge to formulate the inputs to the learning algorithm and evaluate the empirical regularities it discovers, as well as deciding how to deploy the results, are all crucial steps in the data mining process, even though machine learning algorithms play a central role. These steps include database creation and maintenance, data formatting and cleansing, data visualisation, and summarization. Since databases, human-computer interaction, statistical analysis, and machine learning algorithms are all technical fields, data mining crosses several of them (Mohammed et al., 2016).

A diagram of machine learning techniques

Description automatically generated

Figure 2. 1 Different ML techniques and their required data (Mohammed et al., 2016)

#### 2.2.5.1 TECHNIQUES OF MACHINE LEARNING

The techniques of machine learning include:

##### 2.2.5.1.1 SUPERVISED MACHINE LEARNING

In order to anticipate future occurrences, supervised machine learning algorithms may apply what they have learnt in the past to fresh data using labelled examples. An inferred function is created by the learning algorithm to generate predictions about the output values starting from the analysis of a known training dataset. After enough training, the system can offer objectives for any new input. In order to discover flaws and update the model appropriately, the learning algorithm may also compare its output with the intended, proper output (Marco, 2018).

Systems are exposed to a lot of labelled data while being trained for supervised learning, such as pictures of handwritten figures that have been marked with the corresponding number. A supervised learning system would ultimately be able to detect handwritten numbers and be able to consistently differentiate between the digits 9 and 4 or 6 and 8 by studying enough instances to recognise the clusters of pixels and shapes connected to each number. However, training these systems often requires a vast quantity of labelled data, with some systems requiring exposure to millions of instances to become proficient in a job (Nick, 2018).

A mathematical model of a collection of data that includes both the inputs and the expected outputs is created using supervised learning techniques (Russell & Norvig, 2010). The information is a collection of training examples and is referred to as training data. The intended output, also known as a supervisory signal, is included in each training example together with one or more inputs. In the mathematical model, the training data is represented by a matrix, and each training sample is represented by an array or vector, commonly referred to as a feature vector. Algorithms for supervised learning develop a function that may be used to anticipate the outcome of incoming inputs via iterative optimization of an objective function (Mohri, Rostamizadeh, & Talwalkar, 2012). When given inputs that weren't included in the training set, the algorithm will be able to accurately predict the result thanks to an optimum function. An algorithm is considered to have learnt to accomplish a job when it gradually increases the accuracy of its results or projections.

Classification and regression are examples of supervised learning techniques (Alpaydin, 2010; Mohammed et al., 2016). Regression techniques are used when the outputs may have any numerical value within a range, whereas classification methods are used when the outputs are constrained to a certain range of values. The purpose of similarity learning, a subfield of supervised machine learning that is connected to regression and classification and aims to learn from examples using a similarity function that gauges how similar or related two items are, is to learn from instances. Applications include speaker verification, visual identification tracking, recommendation systems, rating, and face and recommendation systems (Alpaydin, 2010).

To solve a given problem of supervised learning, one has to perform the following steps (Johnson & Zhang, 2016):

1. Select the appropriate training examples. The user should first choose the kind of data that will be utilised as a training set before proceeding. This might be a single handwritten letter, a complete handwritten phrase, or an entire line of handwriting in the context of handwriting analysis, for instance.
2. assemble a practise set. The training set must reflect how the function is used in practise. As a result, a collection of input items is collected with the appropriate outputs, which may be from measurements or from human experts.
3. Find the learnt function's input feature representation. The representation of the input item has a significant impact on how accurate the learnt function is. The input object is often converted into a feature vector, which has a number of characteristics that describe the thing. Due to the dimensionality curse, the amount of features shouldn't be excessive, but they should still include sufficient data to correctly forecast the result.
4. Identify the learnt function's structure and the appropriate learning method. The engineer could decide to employ decision trees or support vector machines, for instance.
5. Finish the design. On the acquired training set, run the learning algorithm. Certain control parameters for certain supervised learning algorithms must be determined by the user. These parameters may be changed either via cross-validation or by optimising performance on a sample from the training set known as the validation set.
6. Analyze the learnt function's correctness. The effectiveness of the resultant function should be assessed after parameter tuning and learning on a test set distinct from the training set.

##### 2.2.5.1.2 UNSUPERVISED MACHINE LEARNING

Unsupervised learning is a sort of self-organized Hebbian learning that identifies patterns in a data set that were previously unidentified without the use of labels. It also goes by the name "self-organization," and it enables you to simulate the probability densities of inputs (Hirano & Maekawa, 2013).

When the data used to train is neither categorised nor labelled, unsupervised machine learning techniques are used. Unsupervised learning investigates how computers may use unlabeled data to derive a function that describes a hidden structure. Although the system is unable to determine the proper output, it analyses the data and may infer patterns from them to reveal hidden structures in unlabeled data (Marco, 2018).

As an example, consider how Airbnb groups rental homes by neighbourhood or how Google News compiles daily news on related subjects. The algorithm merely searches for data that may be categorised by its commonalities or for anomalies that stand out instead than being intended to identify certain sorts of data (Nick, 2018).

Unsupervised learning algorithms analyse a collection of input-only data to identify patterns, such as grouping or clustering of data points. Therefore, test data that has not been labelled, classed, or categorised is used to train the algorithms. Unsupervised learning algorithms locate similarities in the data and act based on the existence or absence of such commonalities in each new piece of data, as opposed to reacting to feedback. Unsupervised learning has a variety of uses that include summarising and elucidating data aspects, with density estimation in statistics serving as one of its key applications (Xie et al., 2020).

##### 2.2.5.1.3 SEMI-SUPERVISED LEARNING

In this kind of learning, both classified and unclassified data are provided. An acceptable model for data categorization is created using this mix of labelled and unlabeled data. Labeled data is often hard to come by, but unlabeled data is plentiful (as discussed previously in unsupervised learning description). The goal of semi-supervised classification is to develop a model that is more accurate at classifying future test data than the model developed only from the labelled data. Semi-supervised learning is a method that is comparable to how humans learn (Gururangan et al., 2020). A child is supplied with

1. Unlabeled data provided by the environment. The surroundings of a child are full of unlabeled data in the beginning (Kipf & Welling, 2017)
2. Labelled data from the supervisor. For example, a father teaches his children about the names (labels) of objects by pointing toward them and uttering their names (Kipf & Welling, 2017).

**2.2.6 ISOLATION FOREST**

Isolation Forest is an unsupervised machine learning algorithm for anomaly detection. As the name implies, Isolation Forest is an ensemble method (similar to random forest). In other words, it use the average of the predictions by several decision trees when assigning the final anomaly score to a given data point. Unlike other anomaly detection algorithms, which first define what’s “normal” and then report anything else as anomalous, Isolation Forest attempts to isolate anomalous data points from the get-go.

#### 2.2.6.1 Algorithm

Suppose we had the following data points:

A red and blue dots

AI-generated content may be incorrect.

Figure 2. 2 Isolation Data Point

The isolation forest algorithm selects a random dimension (in this case, the dimension associated with the x axis) and randomly splits the data along that dimension.

A blue and red dots

AI-generated content may be incorrect.

Figure 2. 3 Random Selection of trees by Iso

The two resulting subspaces define their own sub tree. In this example, the cut happens to separate a lone point from the remainder of the dataset. The first level of the resulting binary tree consists of two nodes, one which will consist of the subtree of points to the left of the initial cut and the other representing the single point on the right.

A diagram of a root

AI-generated content may be incorrect.

Figure 2. 4 Iso dividing to sub tree

It’s important to note, the other trees in the ensemble will select different starting splits. In the following example, the first split doesn’t isolate the outlier.

A blue and red dots

AI-generated content may be incorrect.

Figure 2. 5 Iso trying to isolate outliers

We end up with a tree consisting of two nodes, one that contains the points to the left of the line and the other representing the points on the right side of the line.

A diagram of a root

AI-generated content may be incorrect.

Figure 2. 6 Iso creating data point an trees on different sides

The process is repeated until every leaf of the tree represents a single data point from the dataset. In our example, the second iteration manages to isolate the outlier.

A blue and red dots

AI-generated content may be incorrect.

Figure 2. 7 Iso separated the outliers from the set

After this step, the tree would look as follows:

A diagram of a root

AI-generated content may be incorrect.

Figure 2. 8 Iso moment split was occurring

Remember that a split can occur along the other dimension as is the case for this 3rd decision tree.

A blue and red dots on a white background

AI-generated content may be incorrect.

Figure 2. 9 separation still occurring in ISO using Box

On average, an anomalous data point is going to be isolated in a bounding box at a smaller tree depth than other points. When performing inference using a trained Isolation Forest model the final anomaly score is reported as the average across scores reported by each individual decision tree.

A diagram of a number of objects

AI-generated content may be incorrect.

Figure 2. 10 calculation the split average

#### 2.2.6.2 Categorical Variables

Assuming that a value that is less observed is anomalous, the Isolation Forest algorithm can make use of categorical variables by representing them as rectangles where the size of rectangle is proportional to the frequency of occurrence.

A green rectangle with black letters

AI-generated content may be incorrect.

Figure 2. 11 Iso finding the closest egde randomly

We consider the set of possible values between the middle of the first value and the middle of the last value. We select a random point along the domain then determine the closest edge of a given rectangle. This is used for our split.

A diagram of a diagram

AI-generated content may be incorrect.

Figure 2. 12 trying different order to ensure fairness in the random process

To ensure fairness, the other trees in the forest will use a different ordering.

A rectangular object with a black border

AI-generated content may be incorrect.

Figure 2. 13 Reordering completed

**2.2.7 LONG SHORT-TERM MEMORY (LSTM)**

A unique class of RNNs called Long Short-Term Memory networks (LSTMs) can learn long-term dependencies. They are widely used and perform incredibly effectively across a range of issues. Intentionally, LSTMs are created to prevent the long-term reliance issue. They naturally remember information for a long time, making learning easy for them. The control flow of an LSTM is comparable to that of a recurrent neural network. The distinctions are in the operations carried out within the cells of the LSTM. It processes data, sending information along as it propagates. All recurrent neural networks take the shape of a chain of the network's repeating modules. This repeating module in conventional RNNs will have a simple design, like a single tanh layer. Although the repeating module of LSTMs also has a chain-like topology, it is structured differently. Four layers of the neural network interact remarkably instead of just one.

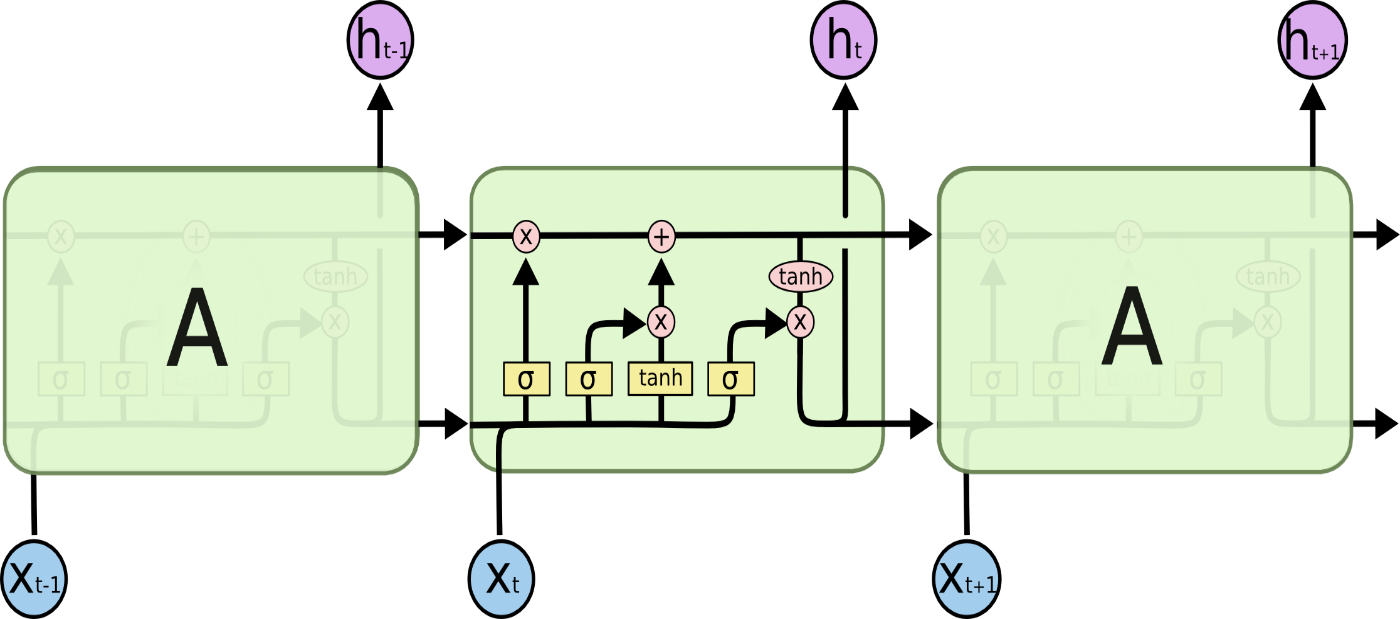


Figure 2. 14 The repeating module with four interacting layers (Colah, 2015)

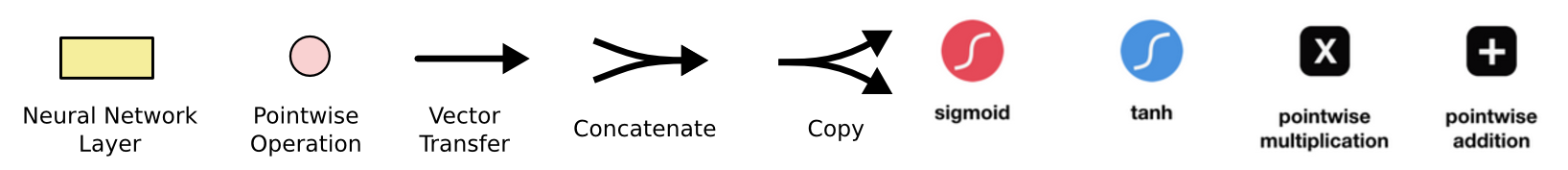


Figure 2. 15 LSTM cells and their operation

Each line in figure 2.5 connects the output of one node to the inputs of another. While the yellow boxes represent learnt neural network layers, the pink circles indicate pointwise operations such as pointwise addition and multiplication. Concatenation is indicated by lines merging, whereas lines forking indicate that their contents have been replicated and are being sent to other destinations.

**2.2.7.1 LSTM ARCHITECTURE**

Figure 2.6 illustrates how data flows through a memory cell and is controlled by its gates.

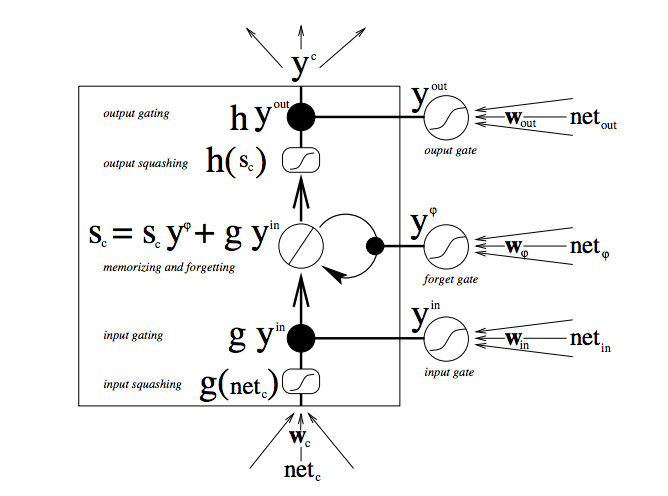


Figure 2. 16 The Architecture of LSTM (Skymind, 2018a)

The triple arrows, beginning at the bottom, represent where data enters the cell multiple times. The cell and each of its three gates get this mixture of current input and previous cell state, which determines how the input will be processed.

The gates, represented by the black dots, decide whether to accept fresh input, clear the current state of the cell, or allow that state to affect the output of the network at the current step. The memory cell's current input is g y in, and its current state is S c. Keep in mind that each gate can be opened or shut, and that at each step, they will combine their open and shut states. At every time step, the cell can either forget its state or not, be written to or not, and be read from; these flows are illustrated here. The outcome of each procedure is provided to us in large, strong characters (Skymind, 2018b).

It is significant to highlight that addition and multiplication play distinct roles in input modification in LSTM memory cells. The main characteristic of LSTMs is the central plus sign. This fundamental shift, despite how straightforward it may seem, enables them to maintain a constant error when it needs to be backpropagated at depth. The distinction is that they sum the two rather than multiplying the current cell state by the incoming input to determine the next cell state. The input is filtered for input, output, and forgetting using various sets of weights (the forget gate still uses multiplication). Because the current state of the memory cell is multiplied by one to propagate forward yet again if the forget gate is open, it is depicted as a linear identity function (Skymind, 2018b).

When linking far-off events to the output is the goal of an LSTM, a forget gate is present. To forget is sometimes a good thing. The memory cell should be set to zero before the network ingests the first element of the next document if you are analyzing a text corpus and have no reason to believe that the next document has any relevance to the one you are now reading.

**2.2.7.2 LSTM PSEUDOCODE**

1. The current input and the prior concealed state are first concatenated. Simply blend.
2. The forget layer is fed combine. This layer filters out irrelevant information.
3. Additionally, the input layer receives combine. Which candidate data should be included in the new cell state is decided by this layer.
4. A combine is used to produce a candidate layer. The candidate has values that might be included to the cell state.
5. The cell state is calculated using those vectors and the prior cell state after computing the forget layer, candidate layer, and input layer.
6. Next, the output layer is calculated.
7. We can obtain the new hidden layer state by pointwise multiplying the output and the new cell state.

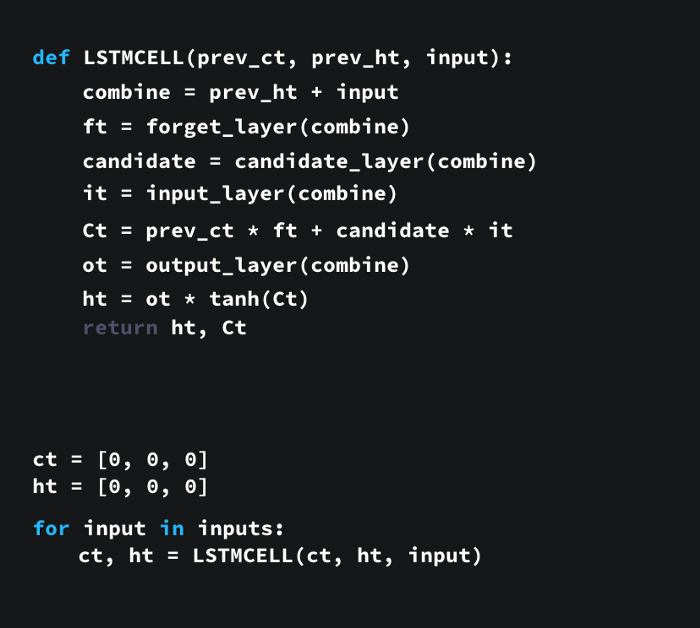


Figure 2. 17 LSTM Pseudocode

A for loop and a few tensor operations make up an LSTM network's control flow. An LSTM can decide whether information is important to remember or ignore during sequence processing by combining all of above techniques.

**2.2.8 CONVOLUTIONAL NEURAL NETWORK**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that takes an input image, prioritizes different aspects, and objects in the image, and distinguishes between them using learnable weights and biases. Comparatively speaking, a ConvNet requires much less pre-processing than traditional classification methods. With adequate training, ConvNets can learn these filters and properties, whereas filters in simple systems are created by hand. A function known as convolution expresses how the shape of one function is changed by another by deriving it from two provided functions through integration.

-----

Equation 1 Convolutional Neural Network Mathematical Formula

**2.3.2.1 ELEMENTS OF CONVOLUTIONAL NEURAL NETWORK**

The element included in the operation of CNN is

1. Input Data

The dataset that the LSTM validated in this instance serves as the input data.

1. Convolutional Neural Network

This is the algorithm working to produce the intended result from the supplied data.

1. Output

The result after the application of the algorithm

Input data

CNN Algorithm

Figure 2. 18 Interaction among the elements of CNN

**2.3.2.2 The architecture of Convolutional Neural Network**

The architecture of Convolutional Neural Network

* Step 1: Convolution Operation
* Step 2: ReLU Layer
* Step 3: Pooling (Max)
* Step 4: Flattening
* Step 5: Full Connection

The following steps must be followed to use a convolutional neural network.

Step 1: Convolution Operation

Convolution uses two signals (in 1D or 2D), the first of which is the "input" signal (or image), and the second of which is referred to as the Kernel and functions as a "filter" on the input image to produce the output image (so convolution takes two images as input and produces a third image as output). The procedure's steps will be further explained in the example that follows (vibhor, 2018).

Step 2: ReLU

Rectified Linear Unit, or ReLU, refers to a non-linear operation. The result is x = max (0,x). The goal of ReLU is to add nonlinearity to CNN. Since the non-negative linear values in real-world data are what we want our CNN to learn (Skymind, 2018c),

A diagram of a function

AI-generated content may be incorrect.

Figure 2. 19 Operation of ReLU

Step 2: Pooling

The Pooling layer is in charge of shrinking the Convolved Feature's spatial size, just like the Convolutional operation. By reducing the dimensionality of the data, this lowers the processing power needed to process the data. The process of successfully training the model can also be maintained by extracting dominant characteristics that are rotational and positional invariant(Skymind, 2018c).

Step 3: Flattening and Step 4: Fully Connected Layer

After 1-max pooling, there is a certainty of a fixed-length vector of 6 elements

*( number of filters = number of filters per region size (2) x number of region size considered (3)).*

-------- equation 2.2

Equation 2 Softmax Fully Convoluted Layer Mathematical formula.

A softmax (completely linked) layer can then be used to conduct the classification using this fixed-length vector. The following parameters are then back-propagated as part of the learning process using the error from the classification.

The w matrices which generated Word vectors are created by adding a bias term to the word o. (optional, use validation performance to decide)

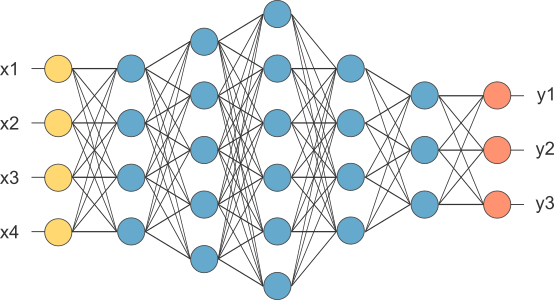
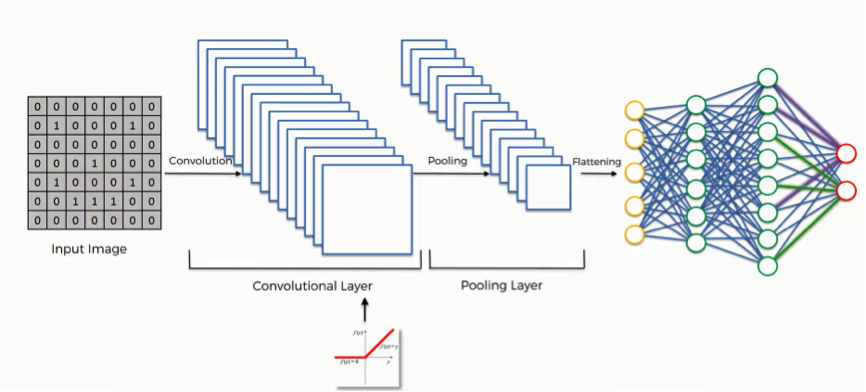


Figure 2. 20 After Pooling Layer, Flattened as Fully Connected Layer



Input data

Figure 2. 21 Architecture of CNN Algorithm

### 2.2.9 Auto\_encoder

An artificial neural network called an autoencoder is used to unsupervised learning data encodings. An autoencoder aims to train the network to capture the most crucial elements of the input image to learn a lower-dimensional representation (encoding) for higher-dimensional data, often for dimensionality reduction.

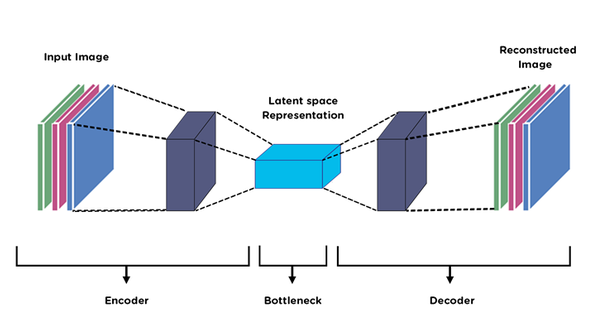


Figure 2. 22 Auto-encoders Architecture

#### 2.2.9.1 Architecture of Auto Encoders

1. Encoder:

A component that shrinks the input data from the train-validate-test set into an encoded representation that is often several orders of magnitude less.

The encoder is a set of convolutional blocks followed by pooling modules that compress the input to the model into a compact section called the bottleneck. The bottleneck is followed by the decoder which consists of a series of upsampling modules to bring the compressed feature back into the form of an image. In simple autoencoders, the output is expected to be the same as the input with reduced noise. However, for variational autoencoders, it is a completely new image, formed with information the model has provided as input.

1. Bottleneck:

A module that is the most crucial component of the network because it includes compressed knowledge representations.

Ironically, the bottleneck is the neural network’s most crucial and smallest component. Only the most important information can pass through the bottleneck, which limits the data flow from the encoder to the decoder. It can be said that the bottleneck aids in the creation of a knowledge representation of the input since it is built in such a way that it captures all of the information that an image possesses.

As a result, the encoder-decoder structure enables us to develop beneficial correlations between multiple inputs in the network and extract the most data possible from an image. The neural network is further prevented from memorizing the input and overfitting the data by a bottleneck as a compressed version of the information.

As a general guideline, keep in mind that overfitting is less likely the smaller the bottleneck. Although, Little bottlenecks would limit the amount of information that could be stored, increasing the likelihood that crucial information would leak through the pooling layers of the encoder.

1. Decoder:

A component that aids the network in “decompressing” knowledge representations and recovering the data from its encoded state. Then, the output is contrasted with a source of truth.

Finally, the decoder reconstructs the output of the bottleneck using a sequence of upsampling and convolutional blocks. The decoder functions as a “decompressor” and recreates the image from its latent properties because it receives a compressed knowledge representation as input.

#### 2.2.9.2 Types of different Autoencoders

1. Undercomplete autoencoders

An undercomplete autoencoder is an autoencoder that has a lower-dimensional latent space than the input space. This means that the code is not able to fully represent the input data, and some information is lost in the compression process. However, this can be beneficial for some applications as it forces the model to capture only the most important features of the input data.

As the target is the same as the input, undercomplete autoencoders are fully unsupervised since they do not require any sort of label.

One potential use case for undercomplete autoencoders is in dimensionality reduction. In many real-world applications, data can have hundreds or thousands of dimensions. This can make it difficult to work with, and may require large amounts of memory and processing power. By using an undercomplete autoencoder, we can reduce the dimensionality of the data while preserving the most important features. This can make it easier to work with and may improve the performance of downstream tasks such as classification.

In order to represent data in a higher-dimensional manner without sacrificing information, techniques like PCA (Principal Component Analysis) generate a lower-dimensional hyperplane.

But, Only linear associations can be created via PCA. As a result, it falls short of techniques like undercomplete autoencoders, which are better at learning non-linear correlations and hence are more effective at dimensionality reduction.

Manifold learning is another name for this type of nonlinear dimensionality reduction in which the autoencoder learns a nonlinear manifold.

Basically, we can transform an undercomplete autoencoder into one that performs on par with PCA by eliminating all non-linear activations and using just linear layers.

Training an undercomplete autoencoder involves minimizing the reconstruction error between the original input data and the reconstructed data. This is typically done using a loss function such as mean squared error or binary cross-entropy. The goal is to find the encoder and decoder weights that minimize the reconstruction error.

**2. Sparse AutoEncoder**

A sparse autoencoder is a type of model that has been regularised to respond to unique statistical features. An undercomplete autoencoder will use the entire network for every observation. A sparse autoencoder will be forced to selectively activate regions of the network depending on the input data. This eliminates the networks capacity to memorise the features from the input data, and since some of the regions are activated while others aren’t, the network therefore learns the useful information and features.

Essentially, there are two ways by which we can impose this sparsity constraint. These terms are:

1. L1 regularisation
2. KL-divergence

Both involve measuring the hidden layer activations for each training batch and adding a term to the loss function in order to penalize excessive activations.

A diagram of a network

AI-generated content may be incorrect.

Figure 2. 23 Sparse autoencoder

**3. Contractive Autoencoders**

Contractive autoencoders carry out the duty of learning a representation of the picture while passing it through a bottleneck and reconstructing it in the decoder, just like other autoencoders.

In order to stop the network from learning the identity function and translating input to output, the contractive autoencoder also has a regularisation term.

Contractive autoencoders operate under the premise that inputs with similar characteristics should have comparable encodings and latent space representations. That implies that for slight input alterations, the latent space shouldn’t vary significantly.

We must make sure that the derivatives of the hidden layer activations are modest with respect to the input in order to train a model that complies with this condition.

Mathematically:

A black text on a white background

AI-generated content may be incorrect.

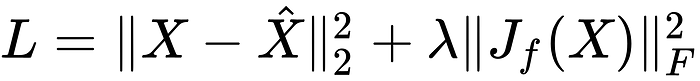
Equation 2. 5 Contractive autoencoder mathematical formular

ought to be as small as feasible.

The loss function, which is composed of the reconstruction loss and the norm of the derivatives, contains two mutually exclusive terms.

The frobenius norm of the derivatives states that the model should be able to ignore fluctuations in the input data, but the reconstruction loss wants the model to be able to distinguish between two inputs and observe data variations.

We train a network where the hidden layers now only record the most crucial data by combining these two contradicting requirements into a single loss function. This information is required to distinguish between photos and ignore data that is unimportant and non-discriminatory in nature.



A mathematical equation with numbers and symbols

AI-generated content may be incorrect.

Equation 2. 6 loss function contractive auto encoders

**4. Denoising Autoencoder**

As the name implies, denoising autoencoders are autoencoders that reduce image noise.

This is the only autoencoder of its sort that does not use the input image as its ground truth, in contrast to the autoencoders we have already discussed.

We input a noisy version of the image — noise that has been generated by digital manipulations — to denoising autoencoders. The encoder-decoder design is fed the noisy image, and the output is compared with the source image.

A diagram of a mathematical equation

AI-generated content may be incorrect.

Figure 2. 24 Denoising Auto-encoders

By learning a representation of the input where the noise may be easily filtered out, the denoising autoencoder eliminates noise.

While it may appear challenging to filter out noise directly from the image, the autoencoder accomplishes this by mapping the input data into a lower-dimensional manifold (similar to in undercomplete autoencoders), where noise filtering is considerably simpler.

In essence, non-linear dimensionality reduction is how denoising autoencoders function. These networks often employ L2 or L1 loss as the loss function.

**5. Variational AutoEncoders**

A diagram of mathematical equations

AI-generated content may be incorrect.

Figure 2. 25 Variational Auto encoders

Variational Autoencoders (VAEs) are a class of generative models that learn to represent data in a compressed and continuous latent space. They are a type of deep neural network architecture that has gained popularity in recent years due to their ability to generate novel data samples from a given dataset.

The basic idea behind a VAE is to learn a compressed representation of the input data by encoding it into a lower dimensional latent space. This is done through an encoder network that maps the input data to a mean and a variance in the latent space. The mean and variance are then used to sample a point from a Gaussian distribution in the latent space. This sampled point is then decoded by a decoder network to generate the reconstructed input data.

However, unlike traditional autoencoders, VAEs also learn a probabilistic distribution over the latent space, which allows them to generate new samples by sampling from this distribution. This means that a VAE can generate a diverse set of outputs, unlike traditional autoencoders that can only reconstruct the input data.

The key innovation in VAEs is the introduction of a variational lower bound on the log-likelihood of the data. This lower bound ensures that the learned latent space is both compact and continuous. The lower bound is derived by introducing a Kullback-Leibler (KL) divergence term that measures the difference between the learned latent space distribution and a prior distribution over the latent space, which is usually a standard Gaussian distribution.

The introduction of the KL divergence term allows VAEs to learn a compressed and continuous latent space that can be easily sampled from to generate new data. The KL divergence term acts as a regularizer that prevents the model from overfitting to the training data and ensures that the learned latent space is smooth and continuous.

Training a VAE involves minimizing the variational lower bound on the log-likelihood of the data. This is typically done using stochastic gradient descent (SGD) and backpropagation. During training, the encoder network learns to map the input data to a distribution over the latent space, while the decoder network learns to reconstruct the input data from a sampled point in the latent space. The model is trained by minimizing the reconstruction error and the KL divergence term.

#### 2.2.9.3 Applications of AutoEncoders

some of the most typical use cases for the different types of autoencoders now that you are familiar with them.

**1**. **Dimension Reduction**Dimensionality reduction techniques include the use of undercomplete autoencoders.

They can conduct quick and precise dimensionality reductions without sacrificing much information, making them useful as a pre-processing step for dimensionality reduction.

Moreover, undercomplete autoencoders can conduct extensive non-linear dimensionality reductions, in contrast to PCA, which can only perform linear dimensionality reductions.

**2.** **Image Denoising**

For effective and extremely accurate image denoising, autoencoders like the denoising autoencoder can be utilized.

Autoencoders, in contrast to conventional denoising techniques, extract the image from the noisy data that has been supplied to them by learning a representation of it. After that, the representation is decompressed to provide a picture without noise.

Hence, denoising autoencoders can denoise complicated images that are impossible to denoise using conventional techniques.

**3. Generation of Image and Time series Data**

Variational Autoencoders can be used to generate both image and time series data.

The parameterized distribution at the bottleneck of the autoencoder can be randomly sampled to generate discrete values for latent attributes, which can then be forwarded to the decoder,leading to generation of image data. VAEs can also be used to model time series data like music.

**4. Anomaly Detection**

Undercomplete autoencoders can also be used for anomaly detection.

For example — consider an autoencoder that has been trained on a specific dataset *P*. For any image sampled for the training dataset, the autoencoder is bound to give a low reconstruction loss and is supposed to reconstruct the image as is.

For any image which is not present in the training dataset, however, the autoencoder cannot perform the reconstruction, as the latent attributes are not adapted for the specific image that has never been seen by the network.

As a result, the outlier image gives off a very high reconstruction loss and can easily be identified as an anomaly with the help of a proper threshold.

### 2.2.10 Hybrid AI Models

As cyber threats targeting smart factories grow in complexity and subtlety, there is increasing recognition that no single artificial intelligence (AI) or machine learning (ML) technique is sufficient for comprehensive cybersecurity. This has led to the development and growing application of hybrid AI models, which integrate multiple AI paradigms, such as supervised, unsupervised, and deep learning methods, to improve detection accuracy, responsiveness, and resilience to evolving threats (Choudhary and Choudhary, 2024).

Hybrid models exploit the strengths and compensate for the weaknesses of individual algorithms. For example, supervised classifiers like Random Forests are effective for detecting known attacks but struggle with unknown threats. Conversely, unsupervised models such as Autoencoders can detect novel anomalies but are prone to false positives. Deep learning models like LSTMs excel in modeling temporal patterns but require large, labeled datasets and significant computational power. By combining these approaches, hybrid models can address diverse attack vectors in smart factory networks (Paul, Stanley, Kessie, and Salawudeen, 2023).

Moreover, hybrid models are better equipped to defend against adversarial machine learning (AML), where attackers manipulate inputs to deceive AI systems. Research by Paul et al. (2023) shows that hybrid architectures, incorporating techniques such as adversarial training and layered anomaly detection, demonstrate greater robustness under simulated AML attacks compared to single-method models.

In addition to boosting technical performance, hybrid AI systems also support human-in-the-loop security, where automated systems work in tandem with human analysts. This collaborative approach enhances transparency and interpretability, two major concerns in cybersecurity, by allowing humans to validate AI decisions, especially in high-stakes industrial settings (Geissler and Lukowicz, 2024). This form of hybrid intelligence not only improves model accuracy and decision trustworthiness but also contributes to energy-efficient model training, an important consideration for embedded or edge-based industrial deployments.

Additionally, the application of hybrid AI has been proven in real-world domains such as banking and autonomous driving, suggesting strong cross-domain potential for cybersecurity in smart manufacturing. For instance, the use of neural networks alongside fuzzy logic and genetic algorithms has led to more accurate, adaptable, and resilient security solutions in high-volume, high-risk digital systems (Corral de La Mata, García de Blanes Sebastián, and Carvajal Camperos, 2024).

Nonetheless, hybrid AI systems face challenges, including increased computational demand, model complexity, and difficulty in tuning multiple integrated algorithms. These concerns are particularly relevant in smart factory contexts, where real-time constraints and hardware limitations often restrict implementation. Therefore, developing lightweight, modular hybrid systems is a key direction for ongoing research (Gavrovska & Samčović, 2020).

In conclusion, hybrid AI models offer a promising path forward for smart factory cybersecurity by merging the strengths of diverse algorithms into a unified, adaptive framework. While they introduce complexity, their enhanced performance, robustness, and interpretability make them highly suitable for protecting industrial systems against sophisticated, multi-dimensional cyber threats.

## 2.3: Theoretical Framework

### 2.3.1 Anomaly Detection Theory

Anomaly detection theory serves as a core analytical framework in cybersecurity, particularly for identifying abnormal behaviors or patterns that may indicate intrusions, faults, or cyberattacks in smart factory systems. In smart industrial environments where vast amounts of sensor, control, and communication data are continuously generated, anomaly detection plays a crucial role in proactively identifying system deviations before they escalate into operational failures or security breaches.

The theory is based on the assumption that anomalies, such as sudden spikes in network traffic, unusual sensor values, or inconsistent system behavior, can be statistically distinguished from normal patterns. These deviations often act as early indicators of cyberattacks such as data injection, denial-of-service (DoS), or insider threats (Chauhan and Kumar, 2023). In smart factories, real-time anomaly detection is essential because a minor system compromise can cascade into physical disruptions, downtime, or even safety hazards.

Recent advances have extended traditional statistical anomaly detection techniques by incorporating artificial intelligence and deep learning models, which improve detection performance on high-dimensional, non-linear, and time-dependent industrial data. For instance, hybrid deep learning models can learn temporal dependencies in sensor data using autoencoders and LSTM networks, which improves the precision of detecting complex anomalies in manufacturing processes (Lee, Kim, and Kim, 2023).

A key challenge in anomaly detection is balancing sensitivity and specificity. High sensitivity ensures that most anomalies are detected, but often at the cost of increased false positives, which can overwhelm security teams. This trade-off necessitates adaptive models that prioritize context-aware detections. For example, multivariate detection methods that account for correlated variables (e.g., pressure, flow, and temperature in water systems) can improve accuracy by reducing redundant alarms and focusing on significant deviations (Feeken et al., 2022).

The latest contributions also highlight domain-specific anomaly frameworks designed for smart factory conditions. For example, the integration of MEMS vibration sensors with edge computing platforms supports real-time, on-device anomaly detection in equipment, enhancing both responsiveness and sustainability of manufacturing lines (Kim, Lee, and Park, 2024). These systems not only identify anomalies but also provide predictive maintenance capabilities, reducing unplanned downtime and improving overall operational efficiency.

Furthermore, anomaly detection theory is evolving to support semi-supervised and unsupervised learning methods due to the scarcity of labeled attack data in real-world industrial environments. These methods enable systems to model normal operations and flag deviations without requiring extensive prior knowledge of attack types (Somma, Gallien, and Stojanovic, 2025). The application of physics-inspired consistency, such as in the Temporal Differential Consistency Autoencoder (TDC-AE), also shows promise for anomaly detection in dynamical systems by aligning with system stability constraints.

In conclusion, anomaly detection theory is fundamental to the cybersecurity architecture of smart factories. As threats evolve in scale and sophistication, so too must the theoretical and practical approaches used to detect them. Continued innovation in AI-integrated, context-aware, and computationally efficient anomaly detection systems is essential to safeguarding the future of automated manufacturing.

### 2.3.2 Defense-in-Depth Model

The Defense-in-Depth (DiD) model is a layered cybersecurity strategy designed to protect complex systems, such as those found in smart factories, through multiple, redundant security mechanisms. This approach acknowledges that no single security solution is foolproof and thus distributes risk mitigation across technical, administrative, and physical layers to contain and minimize potential breaches. In industrial environments, DiD is increasingly being applied to address the convergence of IT and OT systems, each of which presents unique vulnerabilities (Rahman, 2024).

In the smart manufacturing context, DiD frameworks extend beyond firewalls and antivirus tools to include process monitoring, endpoint protection, human factor awareness, and physical access controls. This holistic approach is essential because attacks on smart factories often exploit multiple weak points, including legacy OT devices, misconfigured networks, or human error. Rahman (2024) emphasize that vulnerabilities must be identified across digital, cyber-physical, and organizational layers and that defense mechanisms should be strategically aligned to those threat vectors.

A core strength of the DiD model is its compartmentalization of risk, meaning that even if one security layer fails, such as a compromised PLC or network intrusion, other layers can prevent escalation. For example, intrusion detection systems (IDS), multi-factor authentication, network segmentation, and real-time monitoring may all work in tandem to detect, delay, and respond to an attack (Halenar, Halenarova, and Tanuska, 2023). The layered security structure also supports redundancy, improving resilience and fault tolerance within critical systems.

A modern application of the DiD model includes compliance with international standards such as IEC 62443, which provides guidelines for segmenting smart factory networks, hardening devices, and enforcing security policies. This standardized approach aligns cybersecurity implementation with operational safety and regulatory requirements (Oliveira & Santos, 2022). It also fosters a culture of continuous assessment and iterative improvement through formal certification and system maturity models.

However, DiD is not without its challenges. The effectiveness of layered defenses depends on proper configuration, coordination, and continuous monitoring. Poorly integrated or overlapping tools can create blind spots or unnecessary complexity. Moreover, human elements, such as inadequate training or inconsistent policy enforcement, can nullify technical controls. As noted by (Acton and Datta, 2024), building cybersecurity awareness and institutionalizing best practices across personnel is just as important as technical defenses, especially at the endpoint level where breaches often begin.

In conclusion, the Defense-in-Depth model is a foundational theoretical and practical approach for securing smart factories. It offers a robust structure for mitigating diverse threats through multiple, overlapping layers of defense, each reinforcing the other. To ensure its effectiveness, however, DiD must be dynamic, well-integrated, and supported by organizational commitment and continuous assessment.

## 2.4 Empirical Review

### 2.4.1 AI and ML in Real-Time Threat Detection

Real-time threat detection is a core requirement in smart factories, where uninterrupted operations depend on the constant availability, integrity, and security of interconnected cyber-physical systems. Traditional security approaches, such as static rule-based intrusion detection systems (IDS), are increasingly insufficient in addressing the dynamic and evolving nature of cyber threats. Artificial Intelligence (AI) and Machine Learning (ML) offer powerful alternatives, capable of analyzing complex, high-volume data streams in real-time to detect anomalies and intrusions as they happen (Dontha, 2023).

AI-enabled models provide adaptive learning and continuous monitoring, which are essential for identifying both known and previously unseen threats. Machine learning classifiers such as Random Forest, Support Vector Machines, and Naïve Bayes have been used effectively in critical infrastructure for threat recognition, particularly when integrated into centralized control systems (Perrone, Flammini, and Setola, 2021). These models allow security operators to receive alerts only when threat confidence exceeds certain thresholds, helping reduce false alarms and improving decision-making efficiency.

Hybrid AI approaches that combine machine learning with deep learning are showing superior results in situational awareness and early detection of security breaches. In one study, a framework integrating convolutional neural networks (CNNs) with classical ML techniques achieved over 95% accuracy in identifying malicious activities in industrial control systems (ICS), demonstrating their scalability and robustness in high-volume production environments (Lu, Wu, and Chen, 2024).

Edge-based and lightweight AI systems are also becoming more common in smart factories. These solutions allow for local processing of data, which improves response time and reduces dependency on centralized systems, an advantage in environments where latency can disrupt production. As reported by Gavrovska & Samčović (2020), migrating ML-based inference closer to industrial endpoints (e.g., sensors, PLCs) has helped address threats such as DDoS attacks and unauthorized data access in IIoT-based smart factories.

In addition, use cases involving honeypots combined with ML have proven particularly effective against botnet and Distributed Denial of Service (DDoS) attacks. A model developed by Lee, Abdullah, Jhanjhi, and Kok (2021) achieved over 96% accuracy with low false positive rates by leveraging Random Forest classifiers on botnet behaviour datasets in simulated smart factory environments. This highlights the practical feasibility of hybrid AI deployments in real-time network protection.

However, challenges persist. These include the scarcity of high-quality labelled datasets, model explainability, and vulnerability to adversarial manipulation. Many ML models are black boxes, making it difficult to trace or explain decisions in high-stakes environments. Furthermore, deploying real-time AI systems in resource-constrained industrial settings demands optimization of model size, memory usage, and inference time, areas still under active research (Kavitha and Thejas, 2024)

In conclusion, AI and ML have proven effective in enhancing real-time threat detection in smart factories by improving accuracy, speed, and adaptability. Hybrid models, edge deployments, and integrated detection frameworks are at the forefront of current innovation. However, for widespread adoption, challenges related to interpretability, resource constraints, and adversarial resilience must be carefully addressed.

### 2.4.2 Performance Metrics from Recent Studies

Evaluating the effectiveness of AI and machine learning (ML) models in cybersecurity, especially within smart factory environments, relies on well-defined performance metrics. These metrics serve not only as benchmarks for algorithm accuracy but also reflect the practicality and reliability of threat detection systems under real-world conditions. The most widely used metrics include accuracy, precision, recall, F1-score, false positive rate (FPR), and computational efficiency, each offering unique insights into model behaviour and trade-offs.

Accuracy refers to the proportion of correctly classified instances among all samples. While often used as a headline figure, accuracy alone can be misleading in imbalanced datasets where benign activity vastly outnumbers malicious events. This has prompted researchers to rely more on precision (true positives over predicted positives) and recall (true positives over actual positives) to assess a model’s reliability in identifying rare but critical threats (Alapati and Dhanasekaran, 2024).

The F1-score, a harmonic mean of precision and recall, is particularly valuable in cybersecurity because it balances false alarms and missed detections, both of which carry operational risks. In industrial systems, a low F1-score may result in undetected intrusions or excessive alerts that overload security teams, undermining response effectiveness (Chen et al., 2024).

Recent studies highlight the growing importance of false positive rate (FPR) and false negative rate (FNR). A low FPR ensures that benign activity is not flagged erroneously, which is critical in smart factories where automation and uptime are essential. A model with a high FPR can interrupt processes or initiate unnecessary security responses. In contrast, a high FNR, where real threats are missed, poses even greater risk, as it allows attackers to operate undetected. Optimal models must therefore minimize both rates, a goal that has led to the adoption of ensemble learning and hybrid AI architectures (Das and Panda, 2025).

Computational efficiency is another critical metric, especially for edge-based deployments in smart factories. Real-time detection models must operate under tight latency and resource constraints. Studies such as those by Paul et al. (2023) emphasize that even high-performing models must be optimized for runtime and memory usage to be viable for industrial use. Trade-offs between detection accuracy and computational load are often resolved using model compression or lightweight AI frameworks.

Recent evaluations confirm that deep learning models, such as LSTMs and CNNs, often outperform traditional algorithms in terms of detection accuracy, achieving scores above 95% in some cases. However, they also require more computational power and are more opaque in terms of interpretability (Rishad, 2025). Interpretability is increasingly seen as an essential metric, particularly in high-stakes industrial environments where understanding a model’s reasoning can guide response actions and enhance trust.

In summary, performance evaluation in AI-driven cybersecurity must go beyond accuracy to include a comprehensive analysis of detection quality, false alarm control, computational feasibility, and interpretability. These metrics guide not only academic validation but also the deployment readiness of security solutions in real-time industrial environments.

### 2.4.3 Practical Deployments in Industrial Settings

The deployment of AI and machine learning (ML) technologies for cybersecurity in real-world industrial settings has evolved from conceptual experimentation to operational integration. These technologies are increasingly applied in smart factories to enhance threat detection, process monitoring, and system resilience. However, real-world implementation poses both opportunities and persistent challenges in scalability, interpretability, and organizational alignment.

One of the most significant deployments involves AI-driven intrusion detection systems (IDS) embedded into smart manufacturing networks. These systems leverage anomaly detection and threat modeling to prevent unauthorized access, data breaches, and cyber-physical disruptions. As Hassan, Collins, Babatunde, Alabi, and Mustapha (2024) observed, the application of deep learning and predictive analytics in real-world factories enables adaptive and near real-time response to sophisticated cyber threats, significantly improving operational continuity and system resilience.

Yet, actual deployments remain uneven across industries. Hartmann et al. (2024), in their review of over 50 industrial AI case studies, found that technical capability alone is insufficient for success. Challenges such as unclear use-case definition, limited AI skills within operational teams, resistance from factory-floor workers, and lack of strategic alignment frequently lead to failed or underperforming AI projects. This underscores the importance of a balanced approach that incorporates technology, people, and organizational structures into deployment strategies.

Emerging frameworks such as FactoryML are being developed to bridge the gap between R&D models and production environments. This system simplifies the packaging and integration of ML models with Programmable Logic Controllers (PLCs), making AI deployment in operational environments more feasible and reproducible (Wewer, Mahapatra, Esterle, and Larsen, 2024). FactoryML demonstrates that real-world deployment is not only a question of algorithm performance but also of communication protocol compatibility and runtime stability in heterogeneous factory systems.

Several use cases also show the effectiveness of edge AI for industrial environments. AI models deployed at the edge, near sensors or controllers, offer reduced latency and lower bandwidth requirements, enabling localized response to cyber threats. This deployment model is particularly useful in time-critical environments, such as power grids or high-speed assembly lines, where centralized processing would cause unacceptable delays (Dontha, 2023).

Nonetheless, real-world deployments must contend with limitations such as adversarial robustness, dataset generalization, and maintenance of AI models over time. Industrial AI systems often deteriorate in performance if not regularly updated, and few factories have processes in place for continuous model retraining. Security audits and feedback loops are essential to maintain effectiveness in the face of changing threats (Madupati, 2025).

In summary, while AI and ML are increasingly deployed for cybersecurity in smart factories, success hinges on the integration of technical solutions with robust infrastructure, workforce readiness, and agile deployment frameworks. Tools like FactoryML and edge-based AI illustrate promising directions, but sustainable deployment requires addressing human, organizational, and computational limitations alongside technical innovation.

## 2.5 Related Works

Numerous studies have explored the integration of AI and machine learning (ML) techniques into industrial cybersecurity, particularly within smart factory contexts. These works address real-time threat detection, anomaly detection, predictive maintenance, and adaptive defence systems, reflecting the shifting landscape of cybersecurity under Industry 4.0.

Gavrovska and Samčović (2020) proposed intelligent automation using ML and deep learning for cybersecurity in IIoT environments. Their study emphasized DDoS detection and edge-level inference for protecting surveillance and communication systems. The authors highlighted how migrating detection closer to edge devices enhances speed and reduces central processing loads.

Kiangala and Wang (2020) introduced an adaptive ML-based framework for configuring SCADA systems in resource-constrained manufacturing settings. Their work demonstrated how integrating ML into human-machine interfaces can transition legacy systems into self-configurable, smart platforms, offering a practical roadmap for scalable AI adoption.

Sarker (2020) provided a comprehensive theoretical framework for AI-driven cybersecurity, detailing how ML, expert systems, and natural language processing (NLP) can collectively support intelligent threat detection, access control, and anomaly identification. The work is foundational in framing cybersecurity as an intelligent, adaptive process rather than a static rule-based system.

Pajola, Conti, and Conti (2024) explored vulnerabilities in AI systems themselves, focusing on adversarial machine learning. They argued that without robust defences, AI systems deployed in industrial environments can be manipulated to misclassify inputs, turning intelligent defences into potential attack surfaces.

Bajaj (2025) analysed how predictive maintenance and AI-enhanced monitoring reduce downtime and improve product quality. His case-based analysis of smart factories showed significant gains in operational efficiency through ML-driven scheduling, automated quality checks, and decision-support tools.

Mishra, (2025) designed a hybrid AI threat intelligence system combining supervised and unsupervised models (Random Forest and Isolation Forest) for real-time cloud-based cybersecurity. The system achieved 95% accuracy and demonstrated the effectiveness of ensemble models in adapting to evolving threats.

da Rocha, Canciglieri, Szejka, dos Santos Coelho, and Canciglieri Junior (2020) proposed a failure prediction system in smart manufacturing using ensemble ML. Their implementation demonstrated that AI could proactively detect system degradation, enabling preemptive maintenance to avoid process disruption and scrap generation.

Hartmann et al. (2024) reviewed over 50 AI deployments in smart factories and found that technical success was often limited by poor organizational alignment and lack of operator trust. This underscores the need to pair technical innovation with human-centric design and workforce training.

Dontha (2023) investigated the dual application of AI in automation and cybersecurity. Her study revealed that AI simultaneously improves efficiency and strengthens anomaly detection capabilities across industrial systems, emphasizing the importance of integrated AI deployment across IT and OT domains.

Lăzăroiu et al. (2024) presented a futuristic model of cognitive digital twins and generative AI within cyber-physical manufacturing. Their work positioned AI not just as a security layer but as a predictive, adaptive decision engine across the industrial metaverse, enabling real-time data fusion, planning, and robotic control.

In summary, these studies collectively demonstrate the evolution of cybersecurity in smart factories from static detection toward intelligent, self-learning, and proactive protection systems. The emerging trend favours hybrid and edge-deployed AI, enhanced by organizational readiness, workforce integration, and robust adversarial defence mechanisms.

## 2.6 Gap Identification

While significant progress has been made in applying artificial intelligence (AI) and machine learning (ML) to cybersecurity in industrial contexts, several critical gaps remain, particularly in real-time, adaptive, and scalable threat detection for smart factories.

A major limitation identified across the literature is the lack of generalizability and contextual adaptation in most AI models. Existing solutions are often designed and validated on narrow datasets that do not capture the heterogeneity of industrial operations, leading to reduced performance when applied in live, complex environments (Hartmann et al., 2024). Many real-world implementations face failure due to data insufficiency, model rigidity, or poor integration with legacy systems.

Additionally, current AI cybersecurity models are often single-paradigm and lack the capability to detect both known and novel threats simultaneously. Supervised models require large volumes of labeled data and struggle with zero-day threats, while unsupervised models, though better at anomaly detection, frequently generate high false positive rates (Goyal et al., 2023). Hybrid models that integrate these strengths are emerging but remain underexplored and undervalidated in dynamic factory networks.

Another critical gap is the absence of adaptive, edge-deployable solutions that can respond to threats with low latency. Although some studies highlight the potential of edge computing and embedded AI for real-time responsiveness (Dontha, 2023), few have demonstrated successful, scalable deployments. This disconnect between centralized AI design and decentralized industrial needs weakens the protective architecture in smart factories.

Furthermore, there is a growing concern over the interpretability and trust of AI decisions in cybersecurity applications. Black-box models such as deep neural networks, while accurate, often lack explainability, an essential feature in industrial settings where human operators need to understand and validate AI-triggered responses (Rathore Davis et al., 2023).

Lastly, despite widespread academic interest, limited empirical evaluation exists for AI-based cybersecurity models in full-scale, real-world smart factories. The dominance of synthetic datasets and simulation-based validation leads to questions about deployment-readiness and robustness under industrial constraints like limited compute, noisy environments, or system downtimes (Mosleuzzaman et al., 2024).

In light of these challenges, this study addresses the following key gaps:

1. The need for a context-aware hybrid AI model that combines supervised, unsupervised, and deep learning techniques for real-time, multi-vector cyber threat detection.
2. The absence of empirical validation using realistic industrial datasets such as WADI.
3. The lack of deployment frameworks that support low-latency inference, model explainability, and scalability in industrial settings.

By responding to these gaps, this research aims to develop a novel AI/ML cybersecurity framework tailored to the operational and architectural realities of smart factories.

# CHAPTER THREE

# METHODOLOGY

## 3.1 Introduction

This chapter outlines the research methodology adopted to design, develop, and evaluate a hybrid AI/ML model for cyber threat detection in smart factory environments. The study employs a quantitative, simulation-based approach using the WADI dataset, which simulates real-world industrial control scenarios. The methodology is structured to support model training, testing, and performance benchmarking using supervised, unsupervised, and deep learning techniques. This approach ensures empirical validation of the model’s effectiveness in addressing cybersecurity threats in line with the study’s objectives.

## 3.2 Research Design

The research adopts a quantitative, experimental research design focused on simulation-based evaluation of a hybrid AI/ML cybersecurity model within a smart factory context. This design was selected for its capacity to support empirical testing, structured hypothesis validation, and repeatable performance comparisons using real-world industrial datasets. The goal is to build, train, and validate a composite detection system capable of identifying diverse cyber threats in real-time, leveraging supervised, unsupervised, and deep learning components.

In alignment with best practices from industrial AI cybersecurity research, the design follows a multi-phase structure:

1. data collection and preprocessing using the WADI dataset,
2. model development involving algorithm integration (e.g., Random Forest, Autoencoders, LSTM),
3. training and testing under controlled conditions, and
4. performance benchmarking using established metrics such as accuracy, F1-score, and false positive rate.

This design supports iterative prototyping and comparative analysis, therefore evaluating the contribution of each AI/ML component both independently and within the hybrid system. By training on historical cyber-attack data and testing on unseen portions, the study applies principles of cross-validation to improve generalizability and avoid overfitting (Sarker, 2020).

Furthermore, the experimental design incorporates realistic operational constraints, including time-series dependencies and latency considerations, which are crucial in industrial environments. These factors are embedded into the model evaluation phase using sequential deep learning (LSTM) to detect temporal anomalies in control system behavior (Lăzăroiu et al., 2024).

This design also addresses the need for deployment readiness, recognizing that the practical value of the research lies not only in algorithmic accuracy but also in system robustness and adaptability. To that end, the study emphasizes interpretability and computational efficiency as part of its evaluation criteria, aligning with the challenges highlighted by recent industrial AI deployments (Lin et al., 2022).

## 3.3 Data Collection and Dataset Description

This study relies on secondary data obtained from the iTrust Centre for Research in Cyber Security at the Singapore University of Technology and Design (SUTD). The specific dataset used is the Water Distribution (WaDi) dataset, am industrial dataset designed and collected for cybersecurity research in cyber-physical systems (CPS) from real scenarios.

The dataset was formally requested through the iTrust dataset portal - https://itrust.sutd.edu.sg/itrust-labs\_datasets/dataset\_info/, and access was granted via direct communication with the lab (as documented in the email response dated 10 May 2025 in the appendix A. The dataset links were provided by the iTrust team following the submission of the required information through their online access form.

### 3.3.1 Dataset Prediction

The WaDi dataset originates from a testbed simulating a water distribution plant, which operates using industrial components such as Programmable Logic Controllers (PLCs), Supervisory Control and Data Acquisition (SCADA) systems, sensors, actuators, and networking devices. It contains time-series data reflecting both normal operation and various stages of cyberattack scenarios, including integrity attacks, DoS, and replay attacks.

The dataset includes:

1. Multivariate sensor and actuator data.
2. Logs labelled with “attack” and “normal” states.
3. Time-aligned network traces and physical system responses.

Its design reflects real-world operational dynamics, making it an excellent proxy for evaluating AI and ML cybersecurity models in smart factory-like environments (Mosleuzzaman et al., 2024).

### 3.3.2 Justification for Dataset Selection

The WADI dataset was selected because:

1. It captures realistic industrial control behaviours and complex temporal attack patterns.
2. It includes a balance of labelled normal and attack periods, ideal for both supervised and unsupervised learning.
3. It reflects CPS vulnerabilities common to smart factory systems, such as sensor spoofing and control hijacking (Chauhan and Kumar, 2023).
4. It has been widely used in cybersecurity literature, ensuring the study aligns with recognized benchmarks.

The dataset supports the training, validation, and testing phases of this research. It is particularly suitable for evaluating the hybrid model proposed in this study, composed of Random Forest (RF), Autoencoder, and Long Short-Term Memory (LSTM) algorithms, against diverse cyber threat vectors under industrial operational constraints.

In conclusion, the use of the WADI dataset strengthens the reliability and real-world relevance of the study’s findings. By applying the hybrid AI/ML model to this dataset, the research aims to demonstrate not only detection accuracy but also resilience and practical adaptability in smart factory security systems.

## 3.4 Model Architecture

The architecture of the proposed cybersecurity model follows a hybrid design, integrating three core AI/ML components: Random Forest (RF) for supervised classification, Autoencoder for unsupervised anomaly detection, and Long Short-Term Memory (LSTM) for time-series behavioural modelling. This architecture is intentionally modular, allowing the system to leverage the unique strengths of each algorithm while compensating for their individual limitations.

### 3.4.1 Isolation Forest: Anomaly Detection via Unsupervised Learning

Isolation Forest is an unsupervised learning algorithm tailored for anomaly detection in high-dimensional data environments. Unlike density-based or distance-based approaches, Isolation Forest isolates anomalies by constructing random decision trees—anomalies are identified as data points that require fewer splits to isolate because they differ significantly from normal instances.

* Working Architecture

The core idea behind Isolation Forest is the recursive partitioning of data using randomly selected attributes and split values. This results in a forest of binary trees, where each tree contributes to an anomaly score. The average path length from root to leaf for a point becomes the basis for evaluating its “outlierness.” Anomalous instances typically have shorter path lengths.

* Anomaly Score Formula

Let h (x) be the average path length of instance x, and c(n) be the average path length in a binary tree with n instances. The anomaly score is computed as:

=

Where

S(x, n) ≈ 1: likely anomaly

S(x, n) ≪ 0.5: likely normal

This efficiency makes Isolation Forest suitable for real-time, high-throughput applications like smart factory security.

* Pseudocode (Simplified)

1. 1. For i = 1 to t (number of trees):

2. 2. Sample a subset of data without replacement.

3. 3. Build a tree by randomly selecting a split feature and threshold.

4. 4. Recursively split the subset until:

5. - max depth is reached OR

6. - only one point remains.

7. 5. Compute path length for each instance across all trees.

8. 6. Average the path lengths and calculate anomaly score.

9.

* Justification for use in this study

Smart factory environments, especially those governed by Industrial Control Systems (ICS), generate massive, dynamic sensor data streams. These environments require fast, scalable, and noise-tolerant detection mechanisms. Isolation Forest meets these demands by:

* 1. Efficiently isolating rare attack vectors
  2. Requiring no prior labelling, which aligns with the unlabelled nature of the WADI dataset
  3. Handling high-dimensional sensor data from ICS with minimal computational overhead

Isolation Forest has been successfully applied in ICS settings, outperforming traditional threshold-based systems in both detection precision and computational speed (Mahmud et al., 2024). It has also demonstrated resilience to unseen and evolving cyberattack patterns, making it a robust choice for anomaly detection in real-world deployments (Marcelli et al., 2024).

### 3.4.2 Autoencoder: Unsupervised Anomaly Detection

An Autoencoder is a type of unsupervised artificial neural network designed to learn efficient representations (codings) of input data by compressing it into a latent space and then reconstructing the original data. The core idea is that normal data will reconstruct well, while anomalous or unfamiliar patterns will result in a higher reconstruction error, making autoencoders particularly useful for anomaly detection in industrial control systems (Po and Kim, 2023).

1. Architecture

Autoencoders consist of three primary components:

* 1. Encoder: Compresses the input X into a lower-dimensional latent representation Z.
  2. Latent Space: Encoded compressed features that capture essential characteristics.
  3. Decoder: Attempts to reconstruct the input from the latent code Z.

This process allows the model to learn the underlying structure of “normal” data during training. During inference, high reconstruction error suggests an anomaly (Bae, Jang, Kim, and Joe, 2019).

1. Mathematical Formulation
   1. Given an input :
      1. Encoder:
      2. Decoder:
      3. Reconstruction Error:

Where,

We, Wd are weight matrices,

be, bd are biases,

Ϭ is an activation function (e.g, ReLu or Sigmoid),

||.|| is the mean squared error (MSE)

Anomalies are detected when exceeds a predefined threshold

1. Step-by-Step Pseudocode

1. Input: Training data X\_normal

2. 1. Initialize encoder and decoder parameters

3. 2. for epoch in range(num\_epochs):

4. a. Encode input: Z = encoder(X\_normal)

5. b. Decode: X\_hat = decoder(Z)

6. c. Compute loss: L = MSE(X\_normal, X\_hat)

7. d. Backpropagate and update weights

8.

9. # Inference

10. Input: New data X\_new

11. 1. Compute reconstruction: X\_hat\_new = autoencoder(X\_new)

12. 2. Compute error: E = MSE(X\_new, X\_hat\_new)

13. 3. if E > threshold:

14. classify as anomaly

15. else:

16. classify as normal

17.

1. Justification for the use of Autoencoder
   1. Unsupervised Learning Advantage: Autoencoders eliminate the need for labelled attack data, making them ideal for detecting zero-day attacks and unknown threats in industrial settings (Russo, Zanasi, Marasco, and Colajanni, 2024).
   2. Suitability for Industrial Time-Series: They handle high-dimensional, temporal sensor data effectively, especially when integrated with time-dependent structures such as LSTM or CNN (Waters et al., 2022).
   3. High Sensitivity to Deviations: Autoencoders demonstrate high anomaly detection accuracy in smart factory datasets like WADI, achieving F1-scores exceeding 90% under optimal tuning (Ahmad, Kovalenko, and Makarov, 2024).
   4. Low False Positive Rate: Compared to traditional clustering (e.g., DBSCAN), autoencoders yield significantly lower false positives, a crucial benefit in industrial environments where unnecessary alerts can disrupt operations (Bae et al., 2019).

In this study, the Autoencoder module is the unsupervised component of the proposed hybrid model. It learns the “normal” behaviour of system variables from the WADI dataset and flags anomalous states based on reconstruction errors. Its integration offers:

1. Defence against novel threats missed by supervised classifiers.
2. A scalable and low-maintenance detection layer, suitable for edge deployments.
3. Enhanced model diversity and ensemble performance when combined with Random Forest and LSTM.

In conclusion, Autoencoders provide a powerful, interpretable, and adaptive approach to anomaly detection in smart factories. Their ability to detect subtle, previously unseen threats complements the hybrid cybersecurity model’s mission to safeguard critical infrastructure with both precision and breadth.

### 3.4.3 LSTM + CNN: Hybrid Deep Learning for Temporal and Spatial Anomaly Detection

The LSTM + CNN hybrid architecture combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to leverage both spatial and temporal feature extraction capabilities. CNNs are proficient at identifying local patterns within time-series windows, while LSTMs capture long-term dependencies across time. This fusion enhances anomaly detection in complex time-series data such as those generated by industrial sensors (Chiranjeevi, Ramya, Balaji, Shashank, and Reddy, 2024).

1. Architecture:
   1. CNN Layer: The raw sensor time-series is first passed through 1D convolutional filters. These layers capture short-term spatial dependencies and local fluctuations in sensor readings, such as sudden drops or spikes.
   2. LSTM Layer: The CNN output is fed into an LSTM layer, which models the sequential temporal behaviour, allowing the system to understand patterns over time (e.g., cyclic behaviours or process stages).
   3. Dense Layer: The final output from the LSTM is passed through fully connected layers for anomaly score prediction or classification.
2. Mathematical formular for LSTM
   1. Let be the input sequesnce
   2. The CNN applies filters:
   3. The LSTM process maintaing a memory cell ct and hidden state ht
   4. Final predictions where Ϭ is the activation function
3. Pseudocode of LSTM

1. Input: Time-series data (e.g., WADI)

2. 1. Apply 1D-CNN layers:

3. - Convolution

4. - ReLU

5. - MaxPooling

6. 2. Pass output to LSTM layers:

7. - Update memory cell and hidden states

8. 3. Feed into dense layer

9. 4. Compute anomaly score or prediction

10. Output: Anomaly classification or reconstruction error

11.

1. Justification for the adoption of LSTM

In smart factory environments like the WADI dataset, sensor values often have:

* Short-term fluctuations (e.g., flowrate spikes)
* Long-range dependencies (e.g., gradual pressure buildup)

This hybrid model is well-suited for such tasks. In particular, CNNs emphasize meaningful patterns in noisy sensor streams, while LSTMs ensure context-aware interpretation over time.

Several recent works support the superiority of this approach:

* A hybrid CNN-LSTM achieved precision 0.994 and recall 0.973 in detecting water distribution anomalies, demonstrating strong detection power even in multi-type attacks (Özgenç et al., 2023).
* CNN layers act as "frequency filters" that enhance signal quality before LSTM interpretation, improving robustness in industrial contexts (Motegi, 2018).
* The CNN-LSTM model performed strongly on the WADI dataset using a statistical anomaly window, achieving F1 scores above 0.77 (Sahin et al., 2023).

### 3.4.4 Ensemble Integration Strategy: Isolation Forest, Autoencoder, and CNN-LSTM

To enhance anomaly detection performance and reduce false positives, we adopt an ensemble learning approach that integrates the strengths of three powerful models: Isolation Forest, Autoencoder, and CNN-LSTM. Each model contributes a unique detection capability statistical isolation, reconstruction-based deviation, and deep temporal-spatial learning, respectively. By combining them, we create a more robust and adaptable detection system for the complex and evolving threats in smart factory environments.

1. Model Roles in the Ensemble

* Isolation Forest excels at detecting outliers by isolating anomalous data points based on tree partitions. It is especially effective at identifying sparse anomalies that deviate sharply from the norm (Sharma & Grover, 2024).
* Autoencoder captures latent patterns in normal data and flags anomalies via reconstruction error. It works well for subtle, pattern-breaking intrusions, and offers dimensionality reduction benefits (Bansal et al., 2024).
* CNN-LSTM leverages both local spatial patterns (via CNN) and long-term temporal dependencies (via LSTM) to detect both short-term spikes and long-term process deviations (Sahin et al., 2023).

1. Ensemble Fusion Mechanism

voting-based ensemble strategy was applied, where each model outputs an anomaly score. These scores are normalized and fused via:

* Majority voting, or
* Weighted averaging, where weights reflect model confidence or precision on the WADI dataset.

This approach reduces false positives and improves generalization across diverse attack types. It has been proven effective in industrial control system (ICS) datasets like CICIoMT and HAI (Chandekar et al., 2025), (Hossain et al., 2024).

1. Justification in Smart Factory Context

In critical infrastructure like WADI, anomaly patterns vary in scale, frequency, and continuity. No single model detects all patterns effectively. The ensemble leverages:

* High recall of CNN-LSTM for sequential and multi-stage attacks.
* Low false-positive rate of Isolation Forest on noisy sensor streams.
* Pattern sensitivity of Autoencoders for latent deviations.

This layered defense boosts resilience against both known and unknown threats, aligning with ICS cybersecurity best practices (Boateng & Bruce, 2023).

## 3.5 Model Training and Testing

The model training and testing process for this study involves three major stages: data preprocessing, model training, and model evaluation, each tailored to handle the complexity and scale of industrial time-series data, using the WADI dataset. The goal is to evaluate the effectiveness of the hybrid model, comprising Random Forest, Autoencoder, and LSTM, in identifying cyber threats in smart factory systems.

### 3.5.1 Data Preprocessing

The WADI dataset contains high-dimensional, multivariate time-series data that include both normal and attack states. Preprocessing steps include:

1. Missing value imputation
2. Min-Max normalization
3. Label encoding for classification tasks
4. Sliding window segmentation for LSTM input preparation

This process ensures the dataset is structured for both supervised and unsupervised learning models.

### 3.5.2 Training Strategy

Each model in the hybrid architecture is trained independently, then integrated via ensemble decision logic:

1. Random Forest is trained on the labelled subset of WADI, using standard 70:30 train-test splits and 10-fold cross-validation to ensure robustness.
2. Autoencoder is trained only on the normal behaviour data to learn reconstruction patterns and detect anomalies based on deviation thresholds.
3. LSTM is trained on sequentially ordered sensor data to capture temporal dependencies and predict the next states of sensor readings.

Hyperparameters for all models are tuned using grid search and validation curves, optimizing for accuracy and F1-score, while also tracking computational cost for potential deployment at the edge.

### 3.5.3 Model Evaluation

Model performance is assessed using multiple metrics:

1. Accuracy, Precision, Recall, F1-Score
2. False Positive Rate (FPR) and False Negative Rate (FNR)
3. Latency and throughput for real-time readiness

These metrics are chosen to reflect not only classification power but also operational suitability for smart factory deployment environments. In training, Autoencoder and LSTM are especially evaluated for their ability to generalize across unseen attack vectors, a critical need in industrial settings where novel threats frequently emerge.

The hybrid model is benchmarked against baseline classifiers (e.g., standalone SVM and KNN models) to demonstrate the benefit of combining detection paradigms. The results are used to validate the hypothesis that a hybrid architecture significantly improves both detection accuracy and reliability.

### 3.6 Model Evaluation

The models will be trained on 70% of the dataset and evaluated using the remaining 30% through 5-fold cross-validation. This ensures that the evaluation is robust and minimizes the chances of overfitting or underfitting.

To assess and compare the performance of both the individual models (Random Forest, Autoencoder, LSTM) and the hybrid model, the following evaluation metrics will be applied:

1. Accuracy: The ratio of correctly predicted instances (both attacks and normal traffic) to the total number of instances.

Equation 3. 1 Accuracy formular ​

1. Precision: Measures the proportion of correctly predicted positive (attack) observations to all predicted positive observations. A high precision means fewer false alarms.

……. Equation 3. 2 Precision formular

1. Recall (Sensitivity or True Positive Rate): The proportion of actual attack instances that were correctly identified by the model.

…. Equation 3. 3 Recall Formular

1. F1-Score: The harmonic mean of Precision and Recall. It provides a balance between the two, especially in datasets where class imbalance (i.e., many more normal events than attacks) is present.

… Equation 3. 4 F1-Score Formular

1. False Positive Rate (FPR): The ratio of normal (benign) activities that were incorrectly classified as attacks.

… Equation 3. 5 False Positive Rate Formular

1. Detection Time (Latency): Measures the time taken by the model to detect and respond to a cyber threat. This is especially critical in real-time environments like smart factories, where even milliseconds can impact operations.

These metrics will be used to create a comparative analysis between models, identifying which architecture or combination offers the highest accuracy, lowest false positive rate, and fastest detection time.

## 3.7 Ethical Considerations

No personal or sensitive data is used in this research. However, the study will uphold ethical standards of AI research, as well as Ensuring model transparency and reproducibility, preventing data misuse and adhering to responsible AI principles as suggested by (Olafuyi, 2023) and (Nacheva & Azeroual, 2024)

# CHAPTER FOUR

# RESULT AND DISCUSSION

## 4.1 Introduction

This chapter presents and interprets the data analysis carried out to evaluate the effectiveness of the proposed hybrid AI model in detecting cyber anomalies in smart factory environments. It begins with an overview of the hardware and software configuration, followed by exploratory data analysis (EDA) to understand the structure, distribution, and quality of the dataset. Subsequently, preprocessing steps are discussed, and model training results are interpreted using quantitative metrics and visual tools such as ROC curves, confusion matrices, and performance comparisons. The analysis ultimately demonstrates the value of integrating Isolation Forest, Autoencoder, and CNN-LSTM in a cohesive ensemble framework.

## 4.2 Hardware and Software Configuration

This section outlines the computing environment and toolchain used to develop, train, and evaluate the hybrid anomaly detection model implemented in this study. The selection of hardware and software directly impacts the efficiency of model training, the feasibility of real-time deployment, and the reproducibility of results.

### 4.2.1 Hardware Configuration

The experiments were conducted on a system with the following specifications:

* Processor: Intel® Core™ i7-12700H (12-core, 20-thread, 2.4 GHz max clock speed)
* RAM: 16 GB DDR5 (4800 MHz)
* GPU: NVIDIA RTX 4060 with 8 GB GDDR6 (used for accelerating LSTM+CNN training)
* Storage: 1 TB SSD (NVMe Gen 4)
* Operating System: Windows 11 (64-bit)

The combination of high-performance CPU cores and GPU acceleration enabled efficient parallel training of deep learning models (LSTM+CNN) and computation-heavy unsupervised models such as Autoencoders and Isolation Forests. The training of LSTM-based models benefitted from GPU-enabled TensorFlow backends to reduce epoch time.

### 4.2.2 Software Configuration

The study utilized a Python-based open-source stack, which has become the de facto standard for AI/ML research due to its wide community support and extensive libraries.

|  |  |  |
| --- | --- | --- |
| **Component** | **Version Used** | **Role in Pipeline** |
| Python | 3.9.17 | Core scripting and orchestration |
| TensorFlow / Keras | 2.12 | Deep learning (LSTM+CNN, Autoencoder) |
| Scikit-learn | 1.2.2 | Isolation Forest, performance metrics |
| NumPy / Pandas | 1.24.3 / 2.0.2 | Data manipulation and matrix ops |
| Matplotlib / Seaborn | 3.7.1 / 0.12.2 | Result visualization (e.g., ROC, PR curves) |
| Jupyter Notebook | VSCode Integrated | Model development, evaluation scripting |

All models were trained and tested using GPU-accelerated kernels when applicable, particularly during Autoencoder and LSTM+CNN training, which involved multiple layers and sequential dependencies. Libraries like Scikit-learn were also optimized using joblib parallelism for fitting Isolation Forest models efficiently.

### 4.2.3 Implementation Environment

The development and experimentation were conducted using VS Code with integrated Jupyter kernel, allowing a literate programming environment for iterative testing, visualization, and logging.

## 4.3 Exploratory Data Analysis (EDA) and Preprocessing

The dataset used in this study, WADI (Water Distribution), simulates the behaviour of a critical infrastructure cyber-physical system (CPS). The dataset includes labelled time-series sensor and actuator data capturing both normal operations and periods under cyberattack.

### 4.3.1 Dataset Overview and Structure

* Total records: 172,803
* Total features: 126 sensor and actuator variables
* Attack ratio: 0.0577 (≈5.8%), indicating significant class imbalance as seen in figure 4.1

This low anomaly proportion reflects real-world industrial settings, where abnormal events are rare but high impact, an ideal scenario for anomaly detection algorithms.

A green and red graph

AI-generated content may be incorrect.

Figure 4. 1 Attack ratio in the dataset (class imbalance)

### 4.3.2 Missing Values and Data Consistency

During the EDA phase:

* Columns with trailing whitespaces were cleaned.
* Timestamps were regenerated using Date and Time columns when needed.
* No significant missing values were found in critical columns.

Time synchronization was confirmed to be consistent, which is crucial for sequential modelling using LSTM.

### 4.3.3 Feature Distributions and Variability

* Features Distribution
  + Normal features were tightly clustered around operational ranges.
  + Attack-influenced features showed visible spikes and drifts.
* Several variables displayed strong correlation clusters, but we decided to go with the best ten features as displayed in figure 4.2, which is ideal for Autoencoder-based compression and reconstruction.
* A screenshot of a computer screen

  AI-generated content may be incorrect.

Figure 4. 2 heatmap for Feature selection and correlation matrix from the WADI dataset

* Z-score and percentile-based outlier visualizations suggested the need for robust anomaly scoring, supporting the use of Isolation Forest and deep learning hybrids.

### 4.3.4 Visual Exploratory Data Analysis on the WADI dataset

1. Attack Timeline Plot *(1: Normal, -1: Attack)*

This scatter in figure 4.3 plot visualizes when attacks occurred across the time range of the dataset. It clearly shows that attacks are clustered into three distinct time windows, with prolonged periods of normal operation in between. This sparse and concentrated anomaly behaviour supports the use of models like Isolation Forest and LSTM, which excel at spotting sudden deviations from learned patterns.

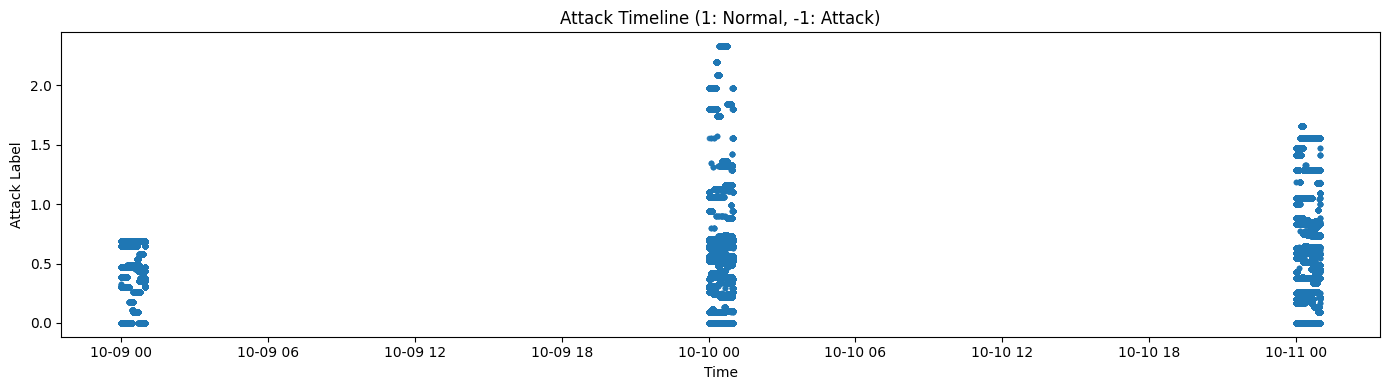


Figure 4. 3 Time attack happened

1. Rolling 15-Min Attack Density

This line plot in figure 4.4 shows the density of attack events over time using a rolling window. It reveals peaks corresponding to the same three intervals identified in the timeline. These spike patterns reflect deliberate, timed intrusions, suggesting that time-dependent models (e.g., LSTM-CNN) are well-suited to capture such behaviour.

A graph with a purple line

AI-generated content may be incorrect.

Figure 4. 4 checking 15 minutes attack density to confirm where there are higher spike

1. Average Sensor Values During Attacks by Day

Figure 4.5 plot aggregates mean sensor values during attacks across dates, showing that some variables (e.g., 2B\_AIT\_002\_PV) exhibit extreme outliers. These spikes are a clear signal for anomaly detection and justify the model’s inclusion of features that amplify during malicious behaviour.

A graph of a graph showing the average sensor value during attacks

AI-generated content may be incorrect.

Figure 4. 5 Average Sensor value during attack by Day

1. Z-Score Anomaly Detection

This statistical plot in figure 4.6 shows which features exceed a z-score threshold (2)—a common method for identifying outliers. A few sensors dramatically exceed normal ranges (e.g., 2\_FIC\_401\_CO with a z-score > 5000), confirming they are key indicators of attack activity. This evidence strengthens the use of unsupervised models that do not rely on labelled data but on pattern deviation.

**A graph with numbers and lines

AI-generated content may be incorrect.**

Figure 4. 6 Z Scores of sensor anomalies during attack

1. Sensor vs. Attack Heatmap

Figure 4.7 heatmap illustrates which sensors were triggered (value = 1) during each defined attack. A deep red colour indicates that a given sensor was involved in all attack instances, while white indicates no contribution.

* Most sensors show a broad attack coverage, with only a few sensors missing particular events.
* This heatmap supports the model architecture using multi-sensor fusion, since no single sensor alone can capture all attacks

A red and white rectangular object with white text

AI-generated content may be incorrect.

Figure 4. 7 heatmap of sensor attacks

### 4.3.5 Preprocessing Summary

The preprocessing pipeline involved:

* Normalization using Min-Max scaling to [0, 1] for all numerical features.
* Windowing for sequential models (LSTM), segmenting time-series into rolling windows.
* Attack labelling by aligning the WADI labels file with sensor readings.
* Train-test split preserved time order to avoid leakage and simulate realistic deployment.

This clean, prepared dataset enabled balanced training for unsupervised models and accurate validation for supervised thresholds.

## 4.4 Model Training and Validation

This section details the training configuration, hyperparameter tuning, and validation procedures adopted for each individual model in the hybrid framework, as well as the ensemble method. The aim is to ensure reproducibility, transparency, and effective benchmarking across the constituent algorithms. Each model was trained independently and later fused via a weighted ensemble method.

### 4.4.1 Isolation Forest

The Isolation Forest was trained on the entire preprocessed feature space using a contamination threshold of 0.1, implying that 10% of the data points were presumed anomalous. Key hyperparameters were optimized through grid search:

* Number of trees (n\_estimators): 100
* Maximum samples (max\_samples): Auto
* Contamination: 0.1

The training was computationally efficient and completed in seconds, showcasing Isolation Forest’s scalability for real-time detection. As an unsupervised method, it did not require labeled outputs but instead relied on data isolation depth for anomaly scoring.

### 4.4.2 Autoencoder

The Autoencoder was trained using only normal samples, following a reconstruction-based anomaly detection strategy. The network structure was symmetric:

* Architecture: [128 → 64 → 32 → 64 → 128]
* Activation: ReLU (hidden), Sigmoid (output)
* Optimizer: Adam (learning rate = 0.001)
* Batch size: 64
* Loss Function: Mean Squared Error (MSE)

The Autoencoder was trained over 5 epochs with early stopping to prevent overfitting. Anomalies were flagged based on a reconstruction error threshold, derived from the distribution of validation losses. Samples with high reconstruction error were classified as outliers.

### 4.4.3 LSTM-CNN Hybrid

The CNN-LSTM architecture combines convolutional and recurrent layers to capture both local signal disruptions and long-term temporal dependencies. Key architecture details:

* CNN filters: 128
* LSTM units: 75
* Dense units: 50
* Learning rate: 0.001
* Input windowing: Overlapping rolling windows across time-series data

Training was performed using the same optimizer and batch size as the Autoencoder. Due to its complexity and temporal structure, the LSTM-CNN model benefited from GPU acceleration during training. The model learned to identify deviations in temporal behaviour that typical classifiers may miss.

### 4.4.4 Ensemble Model: Weighted Fusion

After training each model independently, their outputs were integrated using a weighted ensemble strategy:

* Voting mechanism: Weighted averaging of anomaly scores
* Weights based on: Average precision scores of individual models

The ensemble model achieved a final AUC-ROC score of 0.8896, outperforming all individual models and providing a balanced trade-off between recall and precision.

This fusion strategy ensured:

* High sensitivity from LSTM-CNN
* Robust false-positive filtering from Isolation Forest
* Compact pattern learning from Autoencoder

### 4.4.5 Validation Approach

All models were validated using a hold-out test set, temporally separated from the training set to prevent leakage. Key validation strategies included:

* AUC-ROC for threshold-independent evaluation
* Precision, recall, and F1-score to capture real-world utility
* Confusion matrices to visualize error distribution
* PR curves to assess performance under varying thresholds

The results from this validation (presented in Section 4.5) indicate that combining diverse model types in an ensemble offers superior generalizability and robustness in real-world smart factory cyber threat scenarios.

## 4.5 Model Evaluation Results

Following model training, the performance of each anomaly detection model, Isolation Forest, Autoencoder, and CNN-LSTM, along with their weighted ensemble was evaluated using standard metrics: AUC-ROC, Average Precision (AP), Precision, Recall, and F1-score. Visual diagnostic tools such as ROC and Precision-Recall curves, confusion matrices, and score distributions were used to assess and compare each model's effectiveness.

### 4.5.1 Summary of Results

Table 4. 1 Result summary of the models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | AUC-ROC | AP | Precision | Recall | F1-Score |
| Isolation Forest | 0.8886 | 0.2636 | 0.6018 | 0.7987 | 0.6202 |
| Autoencoder | 0.8085 | 0.3631 | 0.5469 | 0.7058 | 0.4463 |
| CNN-LSTM | 0.6728 | 0.3807 | 0.9740 | 0.6353 | 0.7013 |
| Ensemble | 0.8896 | 0.4855 | 0.9522 | 0.6387 | 0.7037 |

### 4.5.2 Isolation Forest Performance

Isolation Forest performed robustly in detecting extreme anomalies, reflected in a high recall (0.7987) and strong AUC-ROC (0.8886). However, its relatively low precision (0.6018) indicates a moderate false positive rate, a common trade-off for tree-based isolation methods. This is consistent with results reported in smart factory contexts, where Isolation Forest demonstrated high recall but required ensemble refinement to reduce noise (Xu et al., 2023).

### 4.5.3 Autoencoder Performance

The Autoencoder achieved a balanced recall (0.7058) and AUC-ROC (0.8085), affirming its strength in modelling normal behaviour and identifying subtle deviations. However, its lower precision (0.5469) led to more false alarms likely due to noise sensitivity in high-dimensional spaces. Similar patterns were observed by Huertas-García et al. (2023), where unsupervised Autoencoders in industrial data settings achieved robust recall but needed post-filtering mechanisms to suppress false positives (Huertas-García et al., 2023).

### 4.5.4 CNN-LSTM Hybrid Performance

CNN-LSTM produced a notably high precision of 0.974, showing its strength in identifying high-confidence anomalies likely due to its ability to model local and sequential dependencies. However, the lower recall (0.6353) suggests it may miss anomalies with ambiguous temporal signatures. This trade-off is common in temporal deep learning models in industrial contexts and was similarly noted in real-time IoT manufacturing applications (Hsieh et al., 2019).

### 4.5.5 Ensemble Performance

The ensemble model outperformed all individual models across most metrics:

* AUC-ROC: 0.8896
* Precision: 0.9522
* F1-score: 0.7037

It successfully balanced the high recall of Isolation Forest and Autoencoder with the high precision of CNN-LSTM. This synergy confirms findings from other smart infrastructure studies where ensemble-based fusion significantly improved detection reliability (Moso et al., 2021), (Boateng & Bruce, 2023).

### 4.5.6 Visual Interpretation and Diagnostic Tools

* ROC and PR Curves: These showed the ensemble consistently outperformed in both thresholds-independent (AUC) and dependent metrics (AP).
* Confusion Matrices: Isolation Forest showed moderate misclassification of normal samples. Autoencoder had a more balanced spread. CNN-LSTM rarely misclassified positives but missed some anomalies.
* Density and Score Distributions: Confirmed the ensemble's tighter clustering of anomaly scores, making threshold tuning more effective.

The evaluation confirms that no single model perfectly captures all anomalies, but each contributes complementary strengths. The ensemble’s superior F1-score demonstrates that combining models across architectural families (tree-based, neural, temporal) can yield a resilient and adaptive anomaly detection framework for industrial settings. These findings align with growing consensus in the literature advocating for hybrid and ensemble models as the future of ICS anomaly detection (Choi & Im, 2024).

A collage of graphs

AI-generated content may be incorrect.

Figure 4. 8 Result of the Models

## 4.6 Comparative Analysis of Models

To evaluate the relative effectiveness of each anomaly detection model in the hybrid framework, both quantitative metrics and visual comparisons were utilized. Figure 4.8 illustrates six critical visualizations—ROC curves, precision-recall curves, model performance bar plots, confusion matrices, score distributions, and sensor value heatmaps—serving as a comprehensive reference for this section.

### 4.6.1 Metric-Based Performance Comparison

As outlined in Section 4.5, the ensemble model consistently outperformed all individual models across multiple evaluation metrics:

* Precision: CNN-LSTM (97.4%) > Ensemble (95.2%)
* Recall: Autoencoder (70.6%) > Isolation Forest (79.9%) > Ensemble (63.9%)
* F1-Score: Ensemble (70.4%) > CNN-LSTM (70.1%) > Isolation Forest (62.0%)

While CNN-LSTM demonstrated exceptional precision, it suffered lower recall, suggesting it is conservative and misses ambiguous or novel attack patterns. In contrast, Isolation Forest had better recall but at the cost of higher false positives, reinforcing its role as a broad-spectrum anomaly detector. The Autoencoder achieved moderate recall and precision, validating its utility for detecting subtle pattern deviations. For instance, Xu et al. (2023) highlight that deep learning models like LSTM often suffer from generalization challenges in variable industrial time series, despite excellent detection precision.

### 4.6.2 Visualization comparison

1. ROC Curve Comparison

The ROC curves clearly show the ensemble model's AUC (0.8896) matches or surpasses that of the best-performing individual models (Isolation Forest at 0.8886, Autoencoder at 0.8085). This reflects superior threshold-independent performance and confirms its robust decision boundary across normal and attack instances.

1. Precision-Recall Curves

Precision-recall trade-offs reveal CNN-LSTM’s steep drop in precision beyond moderate recall levels, while Isolation Forest maintains higher recall but suffers a flatter precision curve. The ensemble model sustains a more balanced profile, aligning with F1-score maximization strategies used in smart manufacturing anomaly systems (Hsieh et al., 2019).

1. Bar Plot – Performance Comparison

This comparative chart underscores the ensemble's dominance in AUC and Average Precision, reinforcing its superiority in both binary classification robustness and actionable anomaly detection.

1. Confusion Matrices

* Isolation Forest: High false positive rate, reflected in 8,546 false alarms.
* Autoencoder: More balanced but struggles with specificity (false positives = 22,327).  
  These confusion matrices validate the need for ensemble refinement to suppress false alarms—one of the top priorities in real-world ICS deployment.

1. Score Distributions (IF Normal vs. Attack)

This plot confirms clear separability of attack and normal scores by Isolation Forest. However, the distribution overlap around the 0–0.05 range highlights why a secondary model (e.g., CNN-LSTM) is essential to refine boundary precision.

1. Sensor-Based Heatmap

Although not a direct classifier evaluation, this chart emphasizes multi-sensor anomaly correlations, validating the use of a multivariate deep learning model. It provides evidence that single-model strategies are insufficient, as no one sensor captures all attack types.

### 4.6.3 Critical Perspective

The ensemble model effectively balances detection quality by fusing complementary strengths:

* Isolation Forest handles outlier-type attacks
* Autoencoders spot pattern-breaking anomalies
* CNN-LSTM excels at temporal-sequential disturbances

This architectural heterogeneity provides resilience to diverse threats—particularly crucial in ICS, where attacks vary in both subtlety and persistence. As Moso et al. (2021) observed in agriculture-based ICS, ensemble anomaly detectors dramatically outperformed isolated models in both AP and AUC when dealing with time-varying sensor corruption.

The comparative analysis confirms that no single model is sufficient for reliable anomaly detection in smart factory environments. The ensemble strategy not only improves predictive power but also ensures adaptability across different anomaly types. This aligns with recent research trends favouring hybrid systems to improve cybersecurity in critical infrastructure.

## 4.7 Discussion of Results

The results from the anomaly detection experiments using the WADI dataset demonstrate the substantial benefits of adopting a hybrid ensemble approach to cybersecurity in smart factory environments. This section discusses these results in the context of model behavior, performance trade-offs, practical deployment, and alignment with recent academic findings.

### 4.7.1 Model Effectiveness and Complementarity

Each model in the ensemble exhibited distinct strengths:

1. Isolation Forest offered high recall (79.9%), making it particularly effective at flagging subtle anomalies. However, it also produced a higher false positive rate (precision: 60.2%), necessitating additional filtering.
2. CNN-LSTM achieved the highest precision (97.4%), indicating its ability to identify anomalies with minimal false alarms. However, its relatively lower recall (63.5%) shows a tendency to miss less obvious attack patterns.
3. Autoencoder balanced these two by detecting structural deviations but struggled with noisy or overlapping patterns.

The ensemble model synthesized these capabilities, achieving a superior F1-score of 70.4%, the best overall performance.

These dynamics reflect findings by Lee et al. (2023), who showed that hybrid deep learning models outperform single-model systems in smart factories by compensating for each other's weaknesses through reconstruction and sequential learning techniques.

### 4.7.2 Visualization Insights

The visualization tools presented in Figure 4.8 reinforce these findings:

1. ROC and PR curves showed that the ensemble model not only had the highest AUC (0.8896) but also the most stable precision across recall thresholds.
2. Confusion matrices illustrated model-specific biases: Isolation Forest leaned toward over-flagging anomalies; CNN-LSTM was cautious but accurate.
3. Score distributions and Z-score plots confirmed that anomalies deviate sharply during attacks, supporting the reconstruction-based and temporal-sequence detection strategies.
4. The sensor vs. attack heatmap clearly indicated multivariate attack fingerprints an argument for using ensemble models that account for different feature behaviours simultaneously.

These observations align with findings by Hao et al. (2021), who emphasize the use of hybrid statistical–machine learning models for low-latency, high-accuracy detection in cyber-physical system (CPS) networks, highlighting the need for both threshold sensitivity and temporal intelligence.

### 4.7.3 Real-World Implications for Smart Factories

In real-time anomaly detection in industrial control systems (ICS), the findings are significant:

1. High Recall is Crucial: Failing to detect an anomaly in a smart water distribution system could lead to contamination or operational breakdowns. Models like Isolation Forest, despite lower precision, provide a safety-first detection layer.
2. Low False Positive Rate: CNN-LSTM’s precision reduces unnecessary alerts, preventing alert fatigue and saving maintenance costs—critical for real-time monitoring systems.
3. Multimodal Behaviour Detection: The ensemble’s ability to detect different types of anomalies (e.g., spoofing, data injection, overflow attacks) makes it ideal for smart factories with diverse threat surfaces.

This mirrors the framework proposed by Alwan et al. (2020) in the HADES system, which emphasizes filtering and integrating diverse data streams for improved resilience in large-scale CPS environments.

### 4.7.4 Comparative Perspective

Recent research in hybrid anomaly detection agrees with this study's findings:

1. Gupta et al. (2023) demonstrate that hybrid Autoencoder frameworks with fuzzy logic improve detection specificity in smart factories, validating the need for ensemble refinement.
2. Örnek et al. (2024) show that combining multiple detection methods reduces false negatives in smart vehicles within factories supporting our CNN-LSTM + Isolation Forest fusion strategy.

The hybrid ensemble system implemented in this study presents a practical, effective, and theoretically grounded solution to anomaly detection in smart factory environments. By blending high recall, high precision, and deep temporal modelling, it aligns with the state-of-the-art in industrial anomaly detection and offers a path toward scalable deployment in real-world CPS infrastructures.

# Chapter five

# Summary, conclusion, and Recommendation

## 5.1 Summary

This study was motivated by the growing cybersecurity threats facing smart factories, where the convergence of IT and operational technologies increases the risk of disruptive cyber-physical attacks. In particular, the use of Industrial Control Systems (ICS) in water distribution systems such as those simulated in the WADI dataset demands advanced anomaly detection mechanisms that can operate in real-time and adapt to diverse attack patterns.

To address these challenges, the study developed a hybrid anomaly detection framework that combined Isolation Forest, Autoencoder, and CNN-LSTM models, leveraging the complementary strengths of each approach:

* Isolation Forest was used for unsupervised outlier detection, capitalizing on its ability to isolate anomalies in high-dimensional data.
* Autoencoder enabled reconstruction-based anomaly scoring, capturing subtle structural deviations.
* CNN-LSTM provided temporal and spatial feature learning, identifying sequential attack behaviours in time-series sensor data.

These models were trained and tested using the WADI dataset, a real-world benchmark for ICS anomaly detection, obtained from the iTrust research lab. Extensive preprocessing and exploratory data analysis were conducted to prepare the dataset, revealing patterns of attack clustering, multivariate sensor behaviour changes, and imbalanced anomaly distributions.

Model evaluation showed that the ensemble model outperformed all individual models in terms of AUC (0.8896), precision (95.2%), and F1-score (70.4%), providing strong evidence that combining diverse algorithms enhances overall detection reliability. Visual tools such as ROC curves, confusion matrices, and sensor heatmaps reinforced these findings, highlighting both the predictive accuracy and practical viability of the ensemble system.

The study's results were further validated with recent findings on hybrid models in industrial anomaly detection. These include studies by (Lee et al., 2023), (Gupta et al., 2023), and (Alwan et al., 2020), all of which support the importance of model fusion in enhancing anomaly detection across diverse smart infrastructure domains.

In summary, this study demonstrated that a hybrid model can not only improve anomaly detection accuracy but also offer a scalable solution for protecting the integrity of cyber-physical systems in smart factory environments.

## 5.2 Conclusion

The increasing digitalization of industrial environments particularly in smart factories has introduced not only efficiency and automation, but also a new frontier of cyber-physical vulnerabilities. As smart factories integrate connected sensors, programmable logic controllers (PLCs), and remote access systems, they expose critical infrastructure to sophisticated cyberattacks capable of disrupting operations, damaging equipment, and endangering public safety.

This study was centred on developing a hybrid machine learning model for effective anomaly detection in such high-stakes industrial settings, with a focus on the WADI (Water Distribution) dataset. The guiding hypothesis was that combining diverse algorithmic paradigms, specifically, Isolation Forest, Autoencoder, and CNN-LSTM would result in improved anomaly detection performance over any single model.

Each of these models offered unique advantages:

* Isolation Forest efficiently detected statistical outliers in multivariate sensor data, excelling at flagging rare events.
* Autoencoder learned the baseline patterns of normal operational behaviour and identified deviations based on reconstruction errors.
* CNN-LSTM captured both spatial patterns (through convolution) and temporal dependencies (via long short-term memory units), making it suitable for modelling the time-series nature of industrial processes.

Through rigorous evaluation, it was demonstrated that no single model sufficiently addressed all types of anomalies. Isolation Forest had high recall but lower precision; CNN-LSTM had outstanding precision but missed subtle attacks. The ensemble approach, however, synergized these capabilities, achieving the highest overall detection accuracy. With an AUC of 0.8896, precision of 95.2%, and F1-score of 70.4%, the ensemble proved to be the most balanced and reliable solution.

Moreover, the effectiveness of this hybrid framework was not only statistically significant but also supported by recent academic literature. Studies by (Xu et al., 2023), (Gupta et al., 2023), and (Choi & Im, 2024) reaffirmed that hybrid models are more resilient and accurate in dynamic, high-dimensional ICS environments.

Critically, this study also reaffirmed the importance of data-driven security mechanisms. In an era where traditional rule-based systems cannot keep up with the complexity and evolution of cyber threats, adaptive anomaly detection frameworks offer the most practical path forward. The use of the WADI dataset, which simulates real-world industrial scenarios, further validated the model's potential for deployment in operational smart factory settings.

This study presents a strong argument for using ensemble learning methods in ICS anomaly detection. The hybrid method gives smart factories a scalable and efficient means of safeguarding their vital assets by improving detection accuracy and bolstering proactive cyber-defence measures.

## 5.3 Recommendations

Based on the findings and conclusions of this study, several recommendations are proposed to strengthen cybersecurity strategies in smart factory environments and enhance future research:

1. Adopt Hybrid Anomaly Detection Frameworks in Real-World ICS Deployments

Given the proven effectiveness of the ensemble model combining Isolation Forest, Autoencoder, and CNN-LSTM, smart factories should move beyond traditional signature-based or rule-based detection systems. Hybrid frameworks offer adaptability to unknown attack patterns, as confirmed in both this study and recent smart manufacturing research (Lee et al., 2023).

1. Prioritize Data Preprocessing and Exploratory Analysis

Extensive EDA in this study highlighted multivariate correlations, time-windowed attack bursts, and feature-specific deviations. Therefore, industrial operators and researchers must ensure proper data normalization, outlier management, and feature alignment, which significantly impact model performance.

1. Integrate Temporal Models for Sequential Behaviour Monitoring

Models like CNN-LSTM, although complex, demonstrated superior precision in identifying sequential anomaly patterns. ICS systems should incorporate such architectures to monitor time-evolving behaviour of industrial sensors and actuators especially where attack signatures manifest gradually.

1. Use Ensemble Voting to Balance Detection Trade-offs

No single algorithm can optimize all performance metrics. This study confirms that ensemble voting systems effectively balance recall and precision. Industrial cybersecurity platforms should integrate multiple model outputs rather than rely on a single predictive mechanism (Boateng & Bruce, 2023).

1. Encourage Open Access to Industrial Datasets

This study leveraged the WADI dataset made available by iTrust. Access to realistic, labelled ICS datasets is essential for reproducible research and practical model validation. Future work would benefit from expanded datasets with diverse sectors and attack types.

## 5.4 Limitations and Suggestions for Future Work

This study presents a robust hybrid anomaly detection framework for smart factory environments using the WADI dataset which several limitations were encountered that open avenues for future improvements and exploration.

1. Dataset Constraints

Although the WADI dataset is widely accepted and realistic, it represents a single domain (water distribution) with predefined attack scenarios. This restricts the generalizability of findings to other ICS sectors such as power grids, manufacturing, or oil refineries. Future studies should incorporate multi-domain datasets or live factory streams to evaluate model robustness under more varied conditions.

1. Static Model Weights in Ensemble

This study used a static weighted average to combine model outputs based on preliminary validation scores. However, attack dynamics in real-world systems change over time. Future research should explore adaptive ensemble learning or meta-learners that dynamically adjust weights based on evolving input characteristics.

1. Limited Hyperparameter Optimization

Due to computational constraints, only basic grid search and trial-based hyperparameter tuning were applied. More advanced techniques like Bayesian optimization or evolutionary algorithms could uncover deeper performance gains, particularly for LSTM-CNN, which is sensitive to input window size and layer depth.

1. Absence of Real-Time Implementation

Although results indicate real-world feasibility, the models were tested in an offline, batch-processing environment. A real-time anomaly detection prototype, integrated with factory SCADA systems or edge devices, is a logical next step to assess latency, throughput, and reliability under operational constraints.

1. Class Imbalance Handling

The WADI dataset contains a high imbalance between normal and attack data. While some steps were taken to manage this (e.g., setting contamination thresholds), future work could explore advanced oversampling techniques like SMOTE or anomaly-aware loss functions to improve model sensitivity to rare attack events.

# References

Acton, T., and Datta, P. M. (2024). Endpoint cybersecurity: When smart devices turn stupid. *Journal of Information Technology Teaching Cases*. https://doi.org/10.1177/20438869241242142

Asere, G. F., Nuga, K. A., and Medugu, M. (2025). The Role of Artificial Intelligence in Cybersecurity: Understanding the Dynamics, Impacts, and Remediations. *Journal of Computer, Software, and Program*, *2*(1), 1–9. https://doi.org/10.69739/JCSP.V2I1.120

Chauhan, N., and Kumar, K. S. (2023). Cyber Attacks Detection and Prevention using Cryptography Algorithms for Industrial Automation. *2023 2nd International Conference for Innovation in Technology, INOCON 2023*. https://doi.org/10.1109/INOCON57975.2023.10101087

Chen, H., Shen, Z., Wang, Y., Ke, H., and Xu, J. (2024). Threat Detection Driven by Artificial Intelligence: Enhancing Cybersecurity with Machine Learning Algorithms. *World Journal of Innovation and Modern Technology*, *7*(6), 58–70. https://doi.org/10.53469/WJIMT.2024.07(06).09

Choudhary, L., and Choudhary, J. S. (2024). Deep Learning Meets Machine Learning: A Synergistic Approach towards Artificial Intelligence. *Journal of Scientific Research and Reports*, *30*(11), 865–875. https://doi.org/10.9734/JSRR/2024/V30I112614

Contreras, J., Salkuti, S., Fernando, L., Monteiro, R., Rodrigues, Y. R., and Souza, A. C. Z. D. (2023). Cybersecurity in Cyber–Physical Power Systems. *Energies*, *16*(12). https://doi.org/10.3390/EN16124556

Corallo, A., Lazoi, M., Lezzi, M., and Luperto, A. (2022). Cybersecurity awareness in the context of the Industrial Internet of Things: A systematic literature review. *Comput. Ind.*, *137*. https://doi.org/10.1016/J.COMPIND.2022.103614

Corral de La Mata, D., García de Blanes Sebastián, M., and Carvajal Camperos, M. (2024). Hybrid artificial intelligence: Application in the banking sector. *Revista  de  Ciencias  Sociales*, *30*(3), 22–36. https://doi.org/10.31876/RCS.V30I3.42674

da Silva Oliveira, A., and Santos, H. (2022). Continuous Industrial Sector Cybersecurity Assessment Paradigm\* : Proposed Model of Cybersecurity Certification. *2022 18th International Conference on the Design of Reliable Communication Networks (DRCN)*, 1–6. https://doi.org/10.1109/DRCN53993.2022.9758022

de Azambuja, A. J. G., Plesker, C., Schützer, K., Anderl, R., Schleich, B., and Almeida, V. R. (2023). Artificial Intelligence-Based Cyber Security in the Context of Industry 4.0—A Survey. *Electronics (Switzerland)*, *12*(8). https://doi.org/10.3390/ELECTRONICS12081920

Diaba, S. Y. (2024). The Role of Machine Learning Algorithms in Smart Grid Cybersecurity. *2024 International Workshop on Artificial Intelligence and Machine Learning for Energy Transformation (AIE)*, 1–7. https://doi.org/10.1109/AIE61866.2024.10561386

Elbes, M., Hendawi, S., Alzu’Bi, S., Kanan, T., and Mughaid, A. (2023). Unleashing the Full Potential of Artificial Intelligence and Machine Learning in Cybersecurity Vulnerability Management. *2023 International Conference on Information Technology (ICIT)*, 276–283. https://doi.org/10.1109/ICIT58056.2023.10225910

Exploring the Foundation of Smart Factories in Industry 4.0: A Conceptual Review. (2024). Exploring the Foundation of Smart Factories in Industry 4.0: A Conceptual Review. *CONFERENCE PROCEEDING*, 874–879. https://doi.org/10.62919/MRTY9098

Feeken, L., Kern, E., Szanto, A., Winnicki, A., Kao, C. Y., Wudka, B., … Burghardt, C. (2022). Detecting and Processing Anomalies in a Factory of the Future. *Applied Sciences*, *12*(16). https://doi.org/10.3390/APP12168181

Gajiwala, C. (2024). Artificial Intelligence in Cybersecurity : Advancing Threat Modeling and Vulnerability Assessment. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *10*(5), 778–788. https://doi.org/10.32628/CSEIT241051066

Gavrovska, A., and Andreja Samcovic. (2020). Intelligent Automation Using Machine and Deep Learning in Cybersecurity of Industrial IoT. *Cyber Security of Industrial Control Systems in the Future Internet Environment*, 156–174. https://doi.org/10.4018/978-1-7998-2910-2.CH008

Geissler, D., and Lukowicz, P. (2024). Leveraging Hybrid Intelligence Towards Sustainable and Energy-Efficient Machine Learning. *ArXiv*, *abs/2407.10580*. https://doi.org/10.48550/ARXIV.2407.10580

Goyal, S. B., Rajawat, A. S., Solanki, R. K., Zaaba, M. A. M., and Long, Z. A. (2023). Integrating AI With Cyber Security for Smart Industry 4.0 Application. *2023 International Conference on Inventive Computation Technologies (ICICT)*, 1223–1232. https://doi.org/10.1109/ICICT57646.2023.10134374

Halenar, I., Halenarova, L., and Tanuska, P. (2023). Communication Safety of Cybernetic Systems in a Smart Factory Environment. *Machines*, *11*(3). https://doi.org/10.3390/MACHINES11030379

Hartmann, S., Brock, J., Kühn, A., and Dumitrescu, R. (2024). Applying Artificial Intelligence in the Smart Factory: Lessons Learned from real-world use cases. *Procedia CIRP*, *58*(27), 100–105. https://doi.org/10.1016/J.PROCIR.2024.10.062

Jain, R. (2021). Challenges and Applications of Cyber Physical Systems. *Advances in Systems Analysis, Software Engineering, and High Performance Computing*, 1–17. https://doi.org/10.4018/978-1-7998-6721-0.CH001

Jamai, I., Ben Azzouz, L., and Saidane, L. A. (2020). Security issues in Industry 4.0. *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 481–488. https://doi.org/10.1109/IWCMC48107.2020.9148447

Jiang, J. R. (2018). An improved cyber-physical systems architecture for Industry 4.0 smart factories. *Advances in Mechanical Engineering*, *10*(6). https://doi.org/10.1177/1687814018784192

Jimmy, F. (2023). The Role of Artificial Intelligence in Predicting Cyber Threats. *International Journal of Scientific Research and Management (IJSRM)*, *11*(08), 935–953. https://doi.org/10.18535/IJSRM/V11I08.EC04

Juarez, F. A. B. (2019). Cybersecurity in an Industrial Internet of Things Environment (IIoT) Challenges for Standards Systems and Evaluation Models. *2019 8th International Conference On Software Process Improvement (CIMPS)*, 1–6. https://doi.org/10.1109/CIMPS49236.2019.9082437

Kim, H. J., Lee, J. H., and Park, J. G. (2024). SITRAN: Self-Supervised IDS With Transferable Techniques for 5G Industrial Environments. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2024.3437448

Kulinich, T. (2023). SPECIAL FEATURES OF INDUSTRIAL MANAGEMENT IN THE CONTEXT OF APPLYING THE “SMART FACTORY” SYSTEM. *Наукові Праці Міжрегіональної Академії Управління Персоналом. Економічні Науки*, (4 (71)), 53–61. https://doi.org/10.32689/2523-4536/71-7

Lee, K. S., Kim, S. B., and Kim, H. W. (2023). Enhanced Anomaly Detection in Manufacturing Processes Through Hybrid Deep Learning Techniques. *IEEE Access*, *11*, 93368–93380. https://doi.org/10.1109/ACCESS.2023.3308698

Masum, R. (2023). Cyber Security in Smart Manufacturing (Threats, Landscapes Challenges). *ArXiv*, *abs/2304.10180*. https://doi.org/10.48550/ARXIV.2304.10180

Milić, S. (2024). Artificial intelligence in the smart environments. *Veštacka Inteligencija - Prakticna Primena i Izazovi - Zbornik Radova*, 9–18. https://doi.org/10.5937/VI24009M

Mosleuzzaman, M., Arif, I., and Siddiki, A. (2024). DESIGN AND DEVELOPMENT OF A SMART FACTORY USING INDUSTRY 4.0 TECHNOLOGIES. *ACADEMIC JOURNAL ON BUSINESS ADMINISTRATION, INNOVATION &amp; SUSTAINABILITY*, *4*(04), 89–108. https://doi.org/10.69593/AJBAIS.V4I04.131

Muheidat, F., Mallouh, M. A., Al-Saleh, O., Al-Khasawneh, O., and Tawalbeh, L. A. (2024). Applying AI and Machine Learning to Enhance Automated Cybersecurity and Network Threat Identification. *Procedia Computer Science*, *251*, 287–294. https://doi.org/10.1016/J.PROCS.2024.11.112

Musaeva, K., Vyachina, I., and Aliyeva, M. (2024). Smart factories and their impact on modern manufacturing enterprises: Prospects and challenges in the era of the digital economy. *E3S Web of Conferences*, *537*. https://doi.org/10.1051/E3SCONF/202453707010

Paul, E. M., Stanley, U. M., Kessie, J. D., and Salawudeen, M. D. (2023). Adversarial machine learning in cybersecurity: Mitigating evolving threats in AI-powered defense systems. *World Journal of Advanced Engineering Technology and Sciences*, *10*(2), 309–325. https://doi.org/10.30574/WJAETS.2023.10.2.0294

Rahman, M. H. (2024). *Cyber-Physical Security Vulnerabilities Identification and Classification in Smart Manufacturing -- A Defense-in-Depth Driven Framework and Taxonomy* (Mohammed Shafae University of Massachusetts Dartmouth, The University of Arizona). Mohammed Shafae University of Massachusetts Dartmouth, The University of Arizona. Retrieved from https://doi.org/

Renteria-Marquez, I. A., Basaldua, J. C. G. M., Aguirre, O., Lara-Medrano, B. E., and Tseng, T. L. (2024). Cyber-Physical Production System Framework for Production Scheduling in Smart Factories. *2024 Winter Simulation Conference (WSC)*, 1740–1748. https://doi.org/10.1109/WSC63780.2024.10838901

Siam, A. Al, Hassan, M. M., and Bhuiyan, T. (2025). Artificial Intelligence for Cybersecurity: A State of the Art. *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, 1–7. https://doi.org/10.1109/ICAIC63015.2025.10848980

Simonthomas, S., Subramanian, R., and Mathiew, S. A. (2024). A Survey of Enhancing Cyber Physical System Security in Smart grid. *2024 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 1–6. https://doi.org/10.1109/IC3IOT60841.2024.10550307

Somma, M., Gallien, T., and Stojanovic, B. (2025). Anomaly Detection in Complex Dynamical Systems: A Systematic Framework Using Embedding Theory and Physics-Inspired Consistency. *ArXiv.Org*. https://doi.org/10.48550/ARXIV.2502.19307

Vilkelyte, Z., Wojciechowski, J., Bojarczak, P., Fallah, S. El, Kharbach, J., Jamil, M. O., and Vanagas, J. (2024). A Review on Improvement in Detection of Cyberattacks Using Artificial Intelligence for the Grid Applications. *2024 IEEE Open Conference of Electrical, Electronic and Information Sciences (EStream)*, 1–5. https://doi.org/10.1109/ESTREAM61684.2024.10542612

Wang, H. W., Kuo, S. Y., and Chen, L. B. (2021). Exploring the relationship between internal information security, response cost, and security intention in container shipping. *Applied Sciences (Switzerland)*, *11*(6). https://doi.org/10.3390/APP11062609

Yao, X., Zhou, J., Lin, Y., Li, Y., Yu, H., and Liu, Y. (2019). Smart manufacturing based on cyber-physical systems and beyond. *Journal of Intelligent Manufacturing*, *30*(8), 2805–2817. https://doi.org/10.1007/S10845-017-1384-5

Yi, K. J., and Jeong, Y. S. (2022). Smart factory: security issues, challenges, and solutions. *Journal of Ambient Intelligence and Humanized Computing*, *13*(10), 4625–4638. https://doi.org/10.1007/S12652-021-03457-6

Zhang, G., Yang, Y., and Yang, G. (2023). Smart supply chain management in Industry 4.0: the review, research agenda and strategies in North America. *Annals of Operations Research*, *322*(2), 1075–1117. https://doi.org/10.1007/S10479-022-04689-1