information

Review

Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure

Avinash Kumar and Jairo A. Gutierrez *

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, 55 Wellesley Street West, Auckland 1010, New Zealand; git.avinash24@gmail.com
* Correspondence: jairo.gutierrez@aut.ac.nz

Abstract

In the realm of critical infrastructure protection, robust intrusion detection systems (IDSs) are essential for securing essential services. This paper investigates the efficacy of various machine learning algorithms for anomaly detection within critical infrastructure, using the Secure Water Treatment (SWaT) dataset, a comprehensive collection of time-series data from a water treatment testbed, to experiment upon and analyze the findings. The study evaluates supervised learning algorithms alongside unsupervised learning algorithms. The analysis reveals that supervised learning algorithms exhibit exceptional performance with high accuracy and reliability, making them well-suited for handling the diverse and complex nature of anomalies in critical infrastructure. They demonstrate significant capabilities in capturing spatial and temporal variables. Among the unsupervised approaches, valuable insights into anomaly detection are provided without the necessity for labeled data, although they face challenges with higher rates of false positives and negatives. By outlining the benefits and drawbacks of these machine learning algorithms in relation to critical infrastructure, this research advances the field of cybersecurity. It emphasizes the importance of integrating supervised and unsupervised techniques to enhance the resilience of IDSs, ensuring the timely detection and mitigation of potential threats. The findings offer practical guidance for industry professionals on selecting and deploying effective machine learning algorithms in critical infrastructure environments.

Keywords: intrusion detection systems; critical infrastructure

Academic Editors: Jiaping Gui and Futai Zou

Received: 5 May 2025
Revised: 12 June 2025
Accepted: 17 June 2025
Published: 20 June 2025

Citation: Kumar, A.; Gutierrez, J.A. Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure. *Information* **2025**, *16*, 515. https://doi.org/10.3390/info16070515

1. Introduction

Modern human societies operate on an underlying structure of critical infrastructure, which is made up of a wide range of resources and systems that are necessary to maintain essential operations. These infrastructures are crucial to maintaining public safety, economic stability, and national security. They include the complex networks of electricity grids, transportation systems, water treatment plants, and financial institutions. But the same digital technologies that have ushered critical infrastructure sectors into the modern era have also made them a tempting target for malicious cyber actors looking for ways to undermine national security, disrupt operations, or steal sensitive data.

Over the past decade, there has been a notable escalation in cyber threats targeting critical infrastructure sectors worldwide. As noted in [1], the 2015 cyberattack on Ukraine's power grid, widely attributed to state-sponsored actors, stands as a stark example of the vulnerabilities inherent in such systems. This attack resulted in widespread blackouts that affected hundreds of thousands of people and highlighted the potential catastrophic consequences of cyber disruptions to critical infrastructure. Similarly, the Not Petya ransomware attack in 2017 [2] wreaked havoc on several major ports, logistics companies, and financial institutions globally, causing billions of dollars in damages and disrupting operations for long periods.

In this environment of heightened cyber risk and evolving threat landscapes, protecting critical infrastructure systems from cyber intrusions has become of highest importance for governments, industry stakeholders, and cybersecurity professionals alike. Predicting, identifying, and effectively mitigating cyber risks is crucial for preserving the confidentiality, availability, and integrity of vital infrastructure services, as well as reducing the possible effects of cyber incidents on public safety and national security.

Moreover, the ongoing digitization of critical infrastructure systems, coupled with the expansion of associated devices and the Internet of Things, has further increased the difficulties faced by critical infrastructure custodians, as highlighted in this study [3]. These systems' growing interconnectedness and complexity have significantly widened the attack surface, giving attackers numerous ways of entry to take advantage of weaknesses and conduct sophisticated cyberattacks [3]. From remote access points in power grid substations to internet-connected sensors in water treatment plants, the sheer breadth of potential targets presents a formidable challenge for organizations seeking to safeguard critical infrastructure assets from cyber threats.

In response to the escalating cyber threats targeting critical infrastructure, the organizations responsible for their protection have implemented a range of cybersecurity measures to safeguard systems and assets. These measures encompass preventive, detective, and corrective controls aimed at reducing the likelihood and impact of cyber incidents, as per [4]. Among these controls, IDSs are a crucial element of cybersecurity defenses, functioning to monitor network traffic, detect anomalous behavior, and notify security personnel of possible security breaches, among other things.

Traditional IDSs operate on predefined rules, signatures, or heuristics to detect known patterns of malicious behavior, such as known malware signatures or abnormal network traffic patterns. While effective against recognized threats, these IDSs often struggle to detect novel or previously unseen attacks, such as zero-day exploits or polymorphic malware. Additionally, traditional IDSs may generate a high volume of false positives, alerting security personnel to benign activities or legitimate network traffic.

The evolution of cybersecurity threats has prompted the exploration of innovative approaches to enhance the capabilities of IDSs and improve their effectiveness in identifying and reducing risks associated with the internet. One of these strategies is integrating machine learning that allows systems to maximize predictions and extract insights from data into traditional IDSs. By enabling IDSs to detect and identify attacks that were previously not known, reduce false positives, and adapt to emerging threats, machine learning techniques present a promising way to enhance IDS capabilities.

The merger of machine learning with traditional IDSs holds several potential advantages for enhancing cybersecurity defenses. By leveraging machine learning algorithms, IDSs can improve detection accuracy, adaptability, and responsiveness to emerging threats. Dependency on established rules and signatures is reduced because machine learning algorithms can evaluate massive amounts of network data, extract essential details, and identify abnormalities indicative of malicious activity. Furthermore, machine learning gives IDSs the capability to continuously acquire from new data and enhance their models to improve detection accuracy over time, ensuring efficient defense against changing cyberthreats.

Several machine learning algorithms have shown promise in the field of cybersecurity and are potential candidates for integration with traditional IDSs. Algorithms such as

1DCNN and LSTM have been successfully applied to malware detection tasks, as studied by [5], achieving high detection accuracy and low false-positive rates. Unsupervised learning algorithms, such as one-class SVM and isolation forest, give the possibility of identifying anomalies in typical network behavior, which could lead to the detection of unknown or zero-day attacks.

In this paper, we will delve deeper into the cybersecurity challenges facing critical infrastructure sectors and explore the role of machine learning-enabled IDS models in mitigating cyber threats. We will look at the drawbacks of conventional IDSs and talk about how machine learning approaches might help mitigate these drawbacks while refining the effectiveness and resilience of these systems in protecting vital infrastructure assets. Additionally, we will review existing research and case studies on the implementation of machine learning in IDSs for critical infrastructure protection, highlighting key findings and insights.

The present research focuses on evaluating the scalability, flexibility, and generalizability of machine learning-based intrusion detection systems by assessing how well machine learning algorithms detect known and unknown cyber threats using these datasets. Furthermore, by addressing challenges such as data quality, model interpretability, and computational complexity in the context of publicly available datasets, this research aims to provide insights and recommendations that can be applied across various critical infrastructure sectors.

Through empirical studies conducted on publicly available datasets and rigorous experimentation, this research endeavors to aid in the progress of intrusion detection techniques for critical infrastructure protection. Additionally, by focusing on publicly available datasets, this research aims to facilitate transparency, reproducibility, and accessibility in cybersecurity research, thereby enabling broader participation and collaboration within the research community.

Research Questions

Through analysis of the existing literature and careful experimentation in the domain of critical infrastructure datasets and machine learnings algorithms, we look to address the following questions:

- How effective are machine learning algorithms in detecting cyberattacks on critical infrastructure systems?

This question examines how effectively various machine learning algorithms perform in locating and mitigating cyberthreats to vital infrastructure. The study intends to ascertain these algorithm's applicability and reliability in real-world circumstances by examining their accuracy, precision, recall, and general resilience. To protect against ever-evolving cyber threats, machine learning must be effective in fortifying the security posture of vital systems.

- What form of learning algorithms, supervised or unsupervised, are best suited for intrusion detection in critical infrastructure systems?

This question explores the comparative advantages of supervised and unsupervised learning algorithms in the context of intrusion detection for critical infrastructure. The question evaluates how each type of algorithm performs in detecting anomalies and cyberattacks, considering factors such as data availability, anomaly characteristics, and operational efficiency. The study aims to provide a clear understanding of which approach offers superior performance, aiding in the development of more effective security strategies for critical infrastructure protection.

2. IDSs and Machine Learning

An attacker securing illicit access to a device, network, or system is referred to as an intrusion in the context of cybersecurity. Cybercriminals infiltrate organizations covertly by employing ever more advanced methods and strategies. This includes conventional techniques such as coordinated attacks, fragmentation, address spoofing, and pattern deception. By keeping an eye on system logs and network traffic, an IDS is crucial for spotting and stopping these infiltration attempts.

An intrusion detection system (IDS) is a vital part of cybersecurity infrastructure, tasked with monitoring network traffic to detect potential threats and unauthorized activities. IDSs scrutinize system logs and network traffic to identify suspicious patterns or behaviors that could indicate security breaches. Upon detecting an anomaly, the system notifies IT and security teams, enabling them to investigate and address potential security threats, as highlighted in the study by [6]. While some IDS solutions merely report unusual activities, others can proactively respond by blocking malicious traffic. According to [7], IDS products are commonly implemented as software applications on corporate hardware or as part of network security measures. With the rise of cloud computing, numerous cloud-based IDS solutions are available to safeguard a company's systems, data, and resources.

There are various types of IDS solutions, each with their own set of capabilities, as per [8]:

- A Network Intrusion Detection System (NIDS) is positioned at key locations within the network of an organization. It keeps track of all incoming and outgoing traffic, identifying malicious and suspicious activities across all connected devices.
- A Host Intrusion Detection System (HIDS) is deployed on separate devices with internal network and internet connections. It identifies both internal and external threats, such as malware infections and unauthorized access attempts.
- A Signature-based Intrusion Detection System (SIDS) gathers information about known attack signatures from a repository and uses this information to compare observed network packets to a list of known threats.
- An Anomaly-based Intrusion Detection System (AIDS) creates a baseline of typical network behavior and notifies administrators of any deviations from this norm, which may signal potential security breaches.
- A Perimeter Intrusion Detection System (PIDS) is deployed at the boundary of critical infrastructures. It detects intrusion attempts aimed at breaching the network perimeter.
- A Virtual Machine-based Intrusion Detection System (VMIDS) monitors virtual machines to identify attacks and malicious activities within these virtualized environments.
- A Stack-based Intrusion Detection System (SBIDS) is combined into the network layer of the organizational network to analyze packets before they interact with the application layer of the network.

Incorporating machine learning approaches into IDSs has emerged as an intriguing methodology to improve the integrity of critical infrastructure in recent years. Mentioned in [9], the growing sophistication of cyberattacks aimed against vital industries like energy, transportation, and water supply makes conventional rule-based IDSs insufficient in identifying and countering new risks. Additionally, IDSs may more precisely scan enormous amounts of data, spot intricate patterns, and identify legitimate activity through the utilization of machine learning algorithms. IDSs can identify and address attacks in actual time by utilizing machine learning, giving enterprises a proactive defense against intrusions. As a result, there has been an increase in the interest of using machine learning algorithms to complement IDSs, allowing them to adapt to changing threat environments and mitigate newly emerging risks to security.

When it comes to protecting vital infrastructure, IDSs can benefit greatly from machine learning. When compared to conventional rule-based or signature-based approaches, machine learning algorithms are more accurate in detecting small patterns or abnormalities that could be signs of security concerns because they are better at analyzing large amounts of data, as described by [10]. Particularly in anomaly detection, ML techniques can identify previously unknown or zero-day attacks by learning the normal behavior of network traffic or system activities, thus enhancing the IDS's ability to detect novel threats. Moreover, ML algorithms are scalable, adaptable, and capable of efficiently processing large datasets and adapting to changing environments and emerging threats without manual intervention, ensuring continuous protection against evolving cyber threats. ML-based IDSs can operate in near real time, enabling rapid detection and response to security incidents by continuously monitoring network traffic and system logs. Furthermore, ML algorithms can reduce false positives, lowering warning fatigue and allowing security professionals to concentrate on legitimate security issues by examining contextual data and the association between occurrences.

These advantages underscore the effectiveness of ML in enhancing the capabilities of IDSs for critical infrastructure protection, providing a proactive defense against sophisticated cyber threats. Machine learning algorithms offer a wide range of tools and approaches for processing enormous amounts of data and identifying patterns, anomalies, and potential security breaches, as highlighted in the study by [11]. As studied by [12], algorithms can be classified according to their learning methodology and application domain, such as supervised learning, unsupervised learning, and semi-supervised learning approaches.

Supervised Learning: Labeled datasets are used to train these algorithms, with each occurrence attached to a defined class or result. These algorithms are trained to classify novel instances based upon how identical they are to preexisting samples. Supervised learning is a useful feature of IDSs that helps distinguish between malicious and benign network data.

Unsupervised Learning: Algorithms for unsupervised learning analyze unlabeled data, attempting to identify hidden structures or patterns [13]. When it comes to identifying unfamiliar or new dangers that might not be included in training data, these algorithms are especially helpful. Unsupervised learning methods like anomaly detection and clustering can assist IDSs in detecting odd activity that may be a sign of security breaches.

Semi-supervised Learning: By using both labeled and unlabeled data for training, semi-supervised learning incorporates aspects of supervised and unsupervised learning, as per [10]. This method works well in situations where obtaining labeled data is difficult or costly. By harnessing the benefits of both supervised and unsupervised methods, semi-supervised learning in IDSs can raise detection accuracy.

Deep Learning: Raw network traffic or system logs can be analyzed by deep learning models, which can then extract high-level features that point to potential security risks. As mentioned in the study by [14], in IDSs, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are frequently employed for tasks, including malware classification and intrusion detection.

All these approaches to learning from existing data and predicting the outcomes in the new data are suitable and find their applications in varying ways. Not all these approaches are suitable for datasets or environments that vary the nature and amount of data that these algorithms operate upon. Table 1 shows a comparison of machine learning algorithms.

Table 1. Comparison of algorithms.

Algorithm	Learning Type	Strengths	Limitations
Decision Tree	Supervised	Fast, interpretable	Overfits on noisy data
Random Forest	Supervised	Robust to noise, low variance	Slower, less interpretable
LSTM	Supervised	Models long-term temporal dependencies	Needs lots of data, complex tuning
1D CNN	Supervised	Good for pattern recognition in sequences	Low interpretability
Autoencoder	Unsupervised	Learns latent structure of data	Sensitive to noise, tuning-intensive
Isolation Forest	Unsupervised	Efficient with large data	May miss subtle anomalies
One-Class SVM	Semi-supervised	Effective with small attack data	Poor scaling to high-dimensional data

3. Literature Review

Through a systematic assessment of existing literature, we hope to present a thorough overview of the current state of knowledge about the use of machine learning in IDSs for critical infrastructure. By synthesizing findings from many sources, we hope to shed light on the strengths, limitations, and prospective uses of machine learning-based approaches to improving critical infrastructure security.

Our objective in this research is to analyze the existing literature and to place our own work within the larger context of information security and critical infrastructure protection. By critically analyzing the techniques and findings of prior studies, we hope to establish the framework for our own investigation, adding to the ongoing discussion about cybersecurity resilience in an increasingly linked world.

3.1. Background

Systems dealing with critical infrastructure, for example SCADA and other cyber-physical systems, depend heavily on traditional intrusion detection systems to guard themselves against cyberattacks, as noted in the study by [15]. These systems are designed to detect and respond to suspicious activities or anomalies within a network or computing environment.

Traditional IDS architectures typically consist of two main components: sensors and analyzers. Sensors monitor network traffic or system activities and generate alerts upon detecting suspicious patterns or behaviors. Analyzers process these alerts, applying predefined rules or signatures to identify potential threats. Furthermore, a range of detection methods, such as hybrid approaches that combine the two methods, anomaly-based detection, and signature-based detection, can be used in typical IDS designs.

The study by [16] presents insight on how machine learning techniques, both online and offline, can be applied to intrusion detection in cyber–physical systems (CPSs). It analyses the performance of multiple ML techniques using CPS-specific datasets and highlights the importance of a balanced combination of offline and online techniques to enhance CPS security.

Similarly, the work by [17] surveys intrusion detection systems based on machine learning techniques for protecting critical infrastructure. The study discusses the advantages and challenges of employing ensemble models for intrusion detection, emphasizing the need for robust and adaptable detection mechanisms in critical infrastructure.

Additionally, Ref. [18] propose an approach based on a dual-isolation-forest algorithm for anomaly detection in industrial control systems (ICSs), demonstrating improved detection capabilities compared to existing approaches. The framework utilizes two isolation forest models trained on normalized raw data and pre-processed data, achieving enhanced performance in detecting anomalies [18].

Traditional IDSs face several challenges in effectively detecting modern cyber threats. These challenges include the inability to adapt to evolving attack techniques, limited scalability, high false-positive rates, and difficulties in handling encrypted traffic. Moreover, the reliance on predefined signatures or rules makes traditional IDSs susceptible to evasion tactics employed by modern threat actors.

In the realm of cybersecurity, machine learning has become a potent tool, providing novel methods for identifying and reducing a variety of cyber threats. As per [16], the increased frequency of complex cyberattacks has underscored the importance of adopting advanced technologies to enhance the security posture of organizations and critical infrastructures.

A wide range of algorithms and approaches make up machine learning techniques, which enable systems to learn from data and make predictions or judgements, therefore lowering the overhead associated with infrastructure programming, as per the research by [19]. As stated by [10], machine learning techniques are used in cybersecurity for a variety of tasks, such as anomaly detection, malware detection, intrusion detection, and predictive analytics. According to the publication by [17], these methods can analyze enormous volumes of data, spot trends, and identify anomalies that might be signs of security breaches.

In the context of cybersecurity, an evaluation by [20] mentions that ML techniques are applied across various domains, including network security, endpoint security, cloud security, and critical infrastructure protection. These techniques enable organizations to bolster their defense mechanisms, detect previously unknown threats, and proactively mitigate security risks, as reported by [17]. IDSs often rely on predefined signatures or rules to identify known threats, limiting their effectiveness against emerging and sophisticated attacks [16]. Machine learning-based IDSs offer a more dynamic and adaptive approach to threat detection, and are capable of learning from evolving data patterns and adapting to new attack vectors [21].

Network traffic, system logs, and other security-related data sources can all be analyzed by machine learning algorithms to find anomalous activity that might indicate security vulnerabilities, as seen in the paper by [21]. By leveraging advanced analytics and pattern recognition techniques, ML-based IDSs can detect both known and unknown threats, enhancing the overall security posture of organizations.

3.2. Machine Learning Methods for Intrusion Detection

By automatically detecting harmful activity occurring within computer networks, machine learning (ML) techniques are crucial to enhancing the capabilities of intrusion detection systems (IDSs). For classification problems, supervised learning methods like artificial neural networks (ANNs) and support vector machines (SVMs) are frequently used. In these algorithms, the model learns from labeled training data to make predictions on new, unknown data, as mentioned by [22]. For anomaly detection where the model detects deviations from typical behavior in the absence of labeled training data, unsupervised learning techniques, such as clustering and association rule mining, are applied, as demonstrated in the study by [21]. Ref. [18], in their paper, state that semi-supervised and reinforcement learning techniques offer additional capabilities for detecting and responding to evolving cyber threats.

Some of the most commonly used algorithms that can enhance the capabilities of IDSs, according to the existing literature, are listed below:

3.2.1. Supervised Learning Algorithms

Supervised learning algorithms form the bedrock of IDS classification methodologies, leveraging labeled training data to discern normal from malicious network traffic. Support vector machines, as supported by [18], offer robust classification capabilities by delineating optimal hyperplanes to segregate disparate classes of network traffic. Notably, SVMs excel in handling high-dimensional data and have demonstrated efficacy in detecting intrusions within critical infrastructure systems. Decision trees, as explored by [17], exhibit commendable performance in discerning known attacks with remarkable accuracy. However, their efficacy may decrease when faced with novel, previously unseen attack vectors.

Artificial neural networks, studied by [20], show the capacity to discern intricate patterns from network traffic data. Deep neural networks showcase impressive performance in identifying both known and unknown cyber threats. However, for efficient training, they need an extensive amount of labeled training data, as well as processing power.

3.2.2. Unsupervised Learning Algorithms

Algorithms for unsupervised learning, such as autoencoders and k-means clustering, eliminate the need for labeled training data and demonstrate proficiency in identifying abnormal patterns in network traffic.

K-means clustering, as mentioned by [21], clusters network data predicated on similarity metrics, thereby facilitating the detection of outliers diverging from normal behavior. Nevertheless, its efficacy may be curtailed by the exigency to stipulate the number of clusters and its susceptibility to noise.

Autoencoders, as detailed by [23], constitute neural network architectures trained to reconstruct input data. Anomalies are discerned when the reconstruction error surpasses a predefined threshold. Autoencoders adeptly capture complex nonlinear relationships in data but may grapple with elevated false-positive rates in noisy environments.

3.2.3. Semi-Supervised Learning Algorithms

Using both labeled and unlabeled data to improve detection accuracy, semi-supervised learning techniques combine aspects of supervised and unsupervised learning. The one-class SVM, as supported by [23], is an example of a semi-supervised algorithm that constructs an illustration of normal network traffic and flags instances deviating significantly from this representation as anomalies. This approach proves particularly salient in scenarios affected by scarce labeled intrusion data but necessitates the meticulous tuning of hyperparameters to realize optimal performance.

3.2.4. Strengths and Limitations

Each ML technique for intrusion detection offers its unique strengths and grapples with distinct limitations. Supervised learning algorithms excel in ferreting out known attacks, but may falter in the face of zero-day attacks owing to the paucity of labeled training data. Algorithms for unsupervised learning demonstrate effectiveness in detecting abnormalities without any prior knowledge of attack patterns, while they may face high false-positive rates, as stated by [12]. Semi-supervised learning algorithms strike a harmonious balance between supervised and unsupervised approaches but necessitate judicious parameter tuning and may evince sensitivity to the choice of training data.

3.2.5. Traditional IDS Baselines

To contextualize the performance of machine learning-based intrusion detection systems, it is important to consider how traditional IDS approaches operate. These include signature-based detection and threshold-based rule systems. Signature-based IDSs rely on predefined attack signatures, making them highly accurate for known threats but ineffective against zero-day attacks or evolving tactics. Threshold-based systems flag anomalies when metrics exceed preset limits but often suffer from high false-positive rates.

While these methods are computationally efficient and easily interpretable, their lack of adaptability to new or obfuscated threats limits their applicability in dynamic environments such as critical infrastructure. In contrast, ML-based IDSs offer the ability to learn complex patterns, generalize from data, and detect unknown threats, albeit with higher computational costs and reduced interpretability.

ML-based IDSs, while effective at detecting both known and novel threats, are not immune to adversarial manipulation. Attackers may craft inputs to evade detection (evasion attacks) or manipulate training data to degrade model performance (poisoning attacks). These risks are particularly significant in safety-critical environments such as industrial control systems. While this work does not explicitly evaluate adversarial robustness, we recognize it as a key area for future research.

3.2.6. Integration of Machine Learning with Traditional IDSs

Several approaches and frameworks have been proposed to integrate ML with traditional IDSs, aiming to leverage the strengths of both paradigms. Notably, the research by [22] presents a comparative study of AI-based IDS techniques in critical infrastructures. Their study evaluates the performance of ML-driven IDSs in recognizing intrusive behavior in collected traffic data. Similarly, [23] discuss anomaly detection in SCADA systems by the application of one-class classification algorithms, such as SVDD and KPCA, to effectively detect outliers and intrusions.

The integration of ML with traditional IDSs offers several benefits, including improved detection accuracy, adaptability to evolving threats, and scalability. For instance, Ref. [24] demonstrate in their study on real-time threat detection in critical infrastructure that ML models, particularly Logistic Regression, exhibit superior precision and recall in identifying potential hazards. Furthermore, massive data volumes can be handled by ML-based IDSs with efficiency, and they can spot intricate patterns that traditional rule-based techniques could miss.

The main benefit of ML-based IDSs is their capability to improve detection accuracy and adaptability to evolving threats. The research by [18] demonstrates the effectiveness of multiple isolation forest algorithms for anomaly detection in ICSs. Their study indicates superior performance in identifying anomalies and intrusions, thereby enhancing the security of critical infrastructure. Additionally, ML-based IDSs can scale effectively to handle large and diverse datasets, enabling comprehensive threat detection across various network environments.

But there are various difficulties with this integration as well. The computational complexity of machine learning algorithms is one such difficulty, particularly in situations involving real-time detection. Additionally, ensuring the interpretability of ML models remains a concern, as complex models may lack transparency in their decision-making processes. Moreover, adversarial attacks targeting ML-based IDSs pose a significant threat, as highlighted in the research by [21]. Adversaries can manipulate input data to deceive ML models, leading to false alarms or undetected intrusions.

In their research, Ref. [23] discuss the challenges associated with modeling cyberattacks in SCADA systems by applying one-class SVM classification algorithms. These challenges include the need for extensive training data to capture diverse attack scenarios and the optimization of model parameters for robust performance. Moreover, ensuring the interpretability of ML models remains crucial for trust and transparency in decision-making processes. Adversarial attacks targeting ML-based IDSs can exploit vulnerabilities in model architectures, leading to false alarms or undetected intrusions, as highlighted by [21].

3.3. Case Studies and Research Findings

Much research has been undertaken on the efficacy of machine learning-based intrusion detection systems (IDSs) in simulated real-world environments, with an emphasis on identifying cyberthreats in critical infrastructure settings. Ref. [25] conducted a study on real-time threat detection using machine learning and datasets from ICS systems. With an emphasis on accuracy in threat identification, they assessed three machine learning models: K-nearest Neighbors, Random Forest, and Logistic Regression. The outcomes showed that Logistic Regression performed better than the other models, highlighting its importance in strengthening safety controls in critical infrastructure. Specifically, Logistic Regression achieved a precision of 92.5%, recall of 87.3%, and a false-positive rate of 4.1%.

Furthermore, Ref. [25] evaluated the performance of machine learning models in real-time threat detection, focusing on precision, recall, and false-positive rates. Logistic Regression exhibited superior precision and recall compared to other models, emphasizing its balanced approach and high accuracy in predicting potential threats.

Similarly, Ref. [23] investigated anomaly detection in SCADA systems using one-class classification algorithms. Their study focused on the application of Support Vector Data Description (SVDD) and Kernel Principal Component Analysis (KPCA) in detecting anomalies in SCADA systems. The findings showed that these algorithms effectively detected outliers and intrusions, providing a tight description of normal system behavior and enhancing cybersecurity measures in industrial environments. SVDD achieved a precision of 89.6% and a recall of 85.2%, while KPCA achieved a precision of 88.3% and a recall of 82.7%.

In another study, Ref. [18] proposed a dual-isolation-forest-based attack detection framework for industrial control systems (ICSs). Their approach utilized two isolation forest models trained independently using normalized raw data and pre-processed data with Principal Component Analysis (PCA). The framework demonstrated improved performance in detecting attacks, highlighting its significance in ensuring the security of critical infrastructure systems. The proposed framework achieved a detection accuracy of 91.8% on the SWaT dataset and 87.5% on the WADI dataset.

In their comprehensive analysis of anomaly detection algorithms, Ref. [25] compared the performance of sophisticated models, including Interfusion, RANSynCoder, GDN, LSTM-ED, and USAD. The evaluation was based on key metrics, such as precision, recall, false-positive rates, F1 score, and accuracy. The findings indicated variations in model performance across different datasets, with Logistic Regression consistently demonstrating superior performance in accurately identifying potential threats.

Overall, the research findings underscore the effectiveness of machine learning-based IDSs in detecting cyber threats in critical infrastructure environments. The studies highlight the importance of precision and accuracy in threat detection, paving the way for improved security protocols and proactive cybersecurity measures in industrial settings.

3.4. Related Work

This section provides an overview of key studies focusing on the detection of cyber- attacks in industrial control systems (ICSs), particularly utilizing the Secure Water Treatment (SWaT) dataset. Each study contributes unique insights and methodologies, collectively advancing the understanding and combating of cyber threats in critical infrastructure.

Ref. [26] delve into the detection of cyberattacks on water treatment processes, utilizing real data from the Secure Water Treatment (SWaT) testbed. Their research presents a meticulous examination of the challenges posed by cyberattacks on industrial processes, particularly emphasizing the nuances of cyber–physical systems (CPSs) and the intricate communication networks involved. The study offers a comprehensive approach to attack detection, encompassing model-based and data-driven methods tailored to the SWaT process's network architecture and components. By organizing the dataset and analyzing attacks in a structured manner, the study provides valuable insights into designing a monitoring system for SWaT, integrating limit value checks, safety rules, and various monitoring techniques. Moreover, the paper contributes to the field by exploring the complexities of applying model-based monitoring to the SWaT process, addressing challenges such as system complexity and time-varying sensor behavior. Overall, Ref. [26]'s research serves as a significant milestone in understanding and combating cyber threats in critical infrastructure, showcasing the practical application of fault diagnosis techniques and data-driven approaches in industrial settings.

Ref. [27]'s study focuses on the utilization of convolutional neural networks (CNNs) for detecting attacks on critical infrastructure, with a particular emphasis on the SWaT dataset. Their research highlights the efficacy of CNNs in analyzing complex patterns within industrial control environments, surpassing previous methods in anomaly detection tasks. By leveraging the SWaT dataset, the study demonstrates the superiority of CNNs over recurrent networks in detecting abnormal behavior, showcasing CNNs' computational efficiency and performance. The research introduces a statistical window-based anomaly detection method, which successfully predicts future values of data features and measures statistical deviations to identify anomalies. Additionally, the paper highlights how effective various neural network architectures are at detecting anomalies, highlighting CNNs' potential to improve ICS security and dependability. Overall, Ref. [27] provide a solid methodology for identifying cyberattacks in critical infrastructure, which is an important contribution to the area and broadens possibilities for further study on anomaly detection in ICSs utilizing sophisticated neural network architectures.

Ref. [28] introduce the Methodology for Anomaly Detection in Industrial Control Systems (MADICS), aiming to address the lack of standardized methodologies for detecting cyberattacks in ICS scenarios. Their work effectively models ICS behaviors by utilizing deep learning methods in conjunction with semi-supervised anomaly detection. The procedures in MADICS, which include dataset pre-processing, feature filtering, feature extraction, anomaly detection method selection, and validation, are specifically designed to address the difficulties encountered in ICS scenarios. The study surpasses typical performance metrics reported in previous works by achieving state-of-the-art precision, recall, and F1-score values by using MADICS on the SWaT dataset. The research highlights the significance of selecting appropriate hyperparameters and fine-tuning the LSTM model for effective anomaly detection. Moreover, the paper contributes to the discourse on leveraging machine learning, particularly deep learning, for IDSs in critical infrastructure, emphasizing its relevance in addressing security challenges in ICSs. All things considered, the work of [28] offers a thorough approach for identifying cyberattacks in ICSs, setting the stage for further research into boosting the security and dependability of industrial operations.

Ref. [29]'s study focuses on the development of a system-wide anomaly detection approach for industrial control systems (ICSs) using deep learning and correlation analysis. Published in June 2021, the research aims to effectively identify abnormal behavior in industrial control systems critical for various industrial processes. The study introduces a novel technique that leverages deep learning algorithms and correlation analysis to detect anomalies within the system, thereby improving its overall security and reliability. The research emphasizes the multidisciplinary aspect of the method by merging expertise from Queensland University of Technology and Griffith University, with the goal of advancing anomaly detection approaches in ICSs. The suggested method improves the capacity to identify aberrant behavior in industrial control systems using correlation analysis and deep learning algorithms, allowing for prompt intervention and risk-reduction measures. Overall, [29]'s work provides valuable insights into the application of advanced technologies for enhancing security and reliability in critical infrastructure, laying the foundation for future research in anomaly detection in industrial control systems.

3.5. Gaps in the Literature

While the presented literature offers valuable insights into the application of machine learning techniques for intrusion detection in critical infrastructure, several gaps remain unaddressed. Primarily, the existing studies largely focus on evaluating specific algorithms or methodologies in isolation, without a comprehensive comparative analysis that considers the combined performance of both supervised and unsupervised learning approaches. Additionally, while some research highlights the importance of adapting machine learning models to evolving cyber threats, there is limited exploration of frameworks that integrate continuous learning and real-time adaptability within IDSs. Furthermore, the challenges associated with high false-positive rates, particularly in unsupervised learning models, are acknowledged but not thoroughly addressed in terms of practical mitigation strategies. The need for scalable and interpretable models that can handle the complexity and diversity of critical infrastructure environments is also insufficiently explored, leaving a significant gap in developing robust and adaptable IDS solutions. Lastly, there is a lack of extensive case studies or real-world deployments of these advanced machine learning-based IDSs, which would provide practical validation and insights into their effectiveness in operational settings. These gaps underscore the necessity for future research to focus on integrated, adaptive, and scalable machine learning solutions tailored specifically to the dynamic landscape of critical infrastructure cybersecurity.

4. Research Methodology

This section describes the research pathway that will be implemented to explore the answers to our research questions and fulfil the research objectives. We will discuss the complete arc of this research, starting from the literature selection and its analysis to the way we select and work with the proposed dataset, and finally the anomaly detection algorithms and methods used for this research.

4.1. Literature Analysis

In conducting the literature review for this paper, a meticulous methodology was employed to ensure the comprehensive selection of relevant scholarly sources. Leveraging reputable academic databases and platforms such as ResearchGate, Google Scholar, IEEE Xplore, and the library website of Auckland University of Technology (AUT), a systematic search strategy was implemented. This strategy encompassed the use of a diverse set of keywords, including "critical infrastructure cybersecurity", "machine learning in cybersecurity", "intrusion detection systems", "anomaly detection", and others, to capture various facets of the research topic. To maintain relevance and currency, papers published prior to 2012 were excluded, and only those directly related to the protection of critical infrastructure were considered. This stringent inclusion and exclusion criteria ensured that the selected literature aligned closely with the research objectives. The collected papers were organized based on several criteria to facilitate review and analysis. Firstly, papers were categorized according to the specific critical infrastructure they addressed, such as energy, transportation, or healthcare. Furthermore, studies utilizing common datasets, such as the SWaT dataset, were grouped separately for focused analysis. Additionally, papers were segregated based on the type of machine learning algorithms employed, distinguishing between supervised and unsupervised learning approaches. This systematic organization enabled a structured and comprehensive examination of the literature. Throughout the literature search process, several challenges were encountered, including the vast volume of available literature, requiring careful selection and prioritization based on relevance and quality. Moreover, ensuring consistency in search terms and managing duplicate results posed logistical challenges. However, these challenges were effectively mitigated through systematic screening and consultation with the study supervisor to ensure the inclusion of high-quality and pertinent literature.

4.2. Data Collection

In today's highly digitized and interconnected world, facilities that handle operations for critical infrastructure are extremely particular about sharing any data which provides insight into the way these facilities operate, their mechanisms, and any vulnerabilities that the data might conceal. The most widely used datasets for cybersecurity analysis purposes are the DARPA and the NSL-KDD datasets, as highlighted by (Pinto et al., 2023 [17]). The issue with using these datasets for our research was that these datasets are quite old. The DARPA dataset [30] is from 1999, and the NSL-KDD dataset [31] is from 2009; to keep our research abreast with the latest information available in this domain, we decided to use the iTrust SWaT dataset.

The SWaT testbed dataset, sourced from the iTrust [32] website, comprises a comprehensive collection of data, including network traffic logs and readings from all 51 sensors and actuators within the simulated water treatment environment. This dataset encompasses both normal operational periods, spanning 3.5 h, and periods containing six distinct attack scenarios. These attacks were meticulously crafted by the iTrust research team, leveraging sophisticated attack models that consider the intent space of cyber–physical systems (CPSs). The SWaT testbed itself, established in 2015 with funding from the Ministry of Defence (MINDEF) of Singapore and guidance from PUB, Singapore's national water agency, serves as a high-fidelity emulation of a modern water treatment facility. Its construction and design adhere to industry standards, making it a valuable asset for rigorous cyber security research and experimentation. The SWaT dataset used in this study primarily consists of time-series data generated by physical process sensors and actuators embedded in an industrial control system (ICS). It includes readings from multiple devices such as flow indicators, level sensors, motorized valves, and chemical dosing pumps. Notably, the dataset does not contain raw network traffic or packet-level communication data. Therefore, all anomaly detection in this study was performed on controller and sensor/actuator data rather than network-layer telemetry.

This choice aligns with real-world operational priorities in critical infrastructure, where process integrity is often monitored more directly through physical variables rather than abstract network flow. The rationale for focusing on ICS-level telemetry is to identify attacks that manifest through physical anomalies or unauthorized actuator behavior. Access to the SWaT dataset was facilitated through a formal request process via the iTrust website. Upon submission of the request form, access to the dataset was granted, and a download link was provided via the requester's academic email address. It was imperative to adhere to the terms of usage outlined by iTrust, which included giving explicit credit to the organization in any resulting publications and refraining from sharing the dataset with others without explicit permission.

4.3. Data Pre-Processing

This section focuses on preparing the dataset for analysis with several machine learning and deep learning models to identify anomalies. The process was taken directly from the chosen critical infrastructure scenario, and the main goals of this preprocessing step involved subdividing the dataset into training, validation, and test sets; searching the dataset for erroneous or corrupted values; properly handling missing values; encoding categorical features; and scaling continuous features for model training.

The first step is to examine the dataset to find and eliminate erroneous or corrupted entries, which tend to occur during the ICS's warm-up or transitory stages. Plotting each feature individually versus time allows us to visually detect outlier values and remove them from the dataset. Furthermore, appropriate procedures are implemented to detect and handle missing values. The missing values are identified as not valid in order to preserve their contribution to our understanding of cyberattacks, as opposed to being set to the mean or median, or having damaged samples removed.

The second goal is to use one-hot encoding (OHE) to prepare categorical features for machine learning models to enhance learning capabilities, as per [33]. With this encoding technique, binary characteristics are used in place of categorical features to represent each distinct categorical value. Additionally, the features that only show the states of certain sensors in the system are identified, and label encoding is applied to them to make the features more conducive for machine learning algorithms. To improve integration into neural network design, an embedding layer may be used to convert the set of OHE categorical features into a vector of continuous features.

Next, we use standardization or min–max scaling to scale continuous features. Subtracting the mean and dividing by the standard deviation is the standardization process, which works well for data that is regularly distributed. However, by scaling features to a range of [0, 1] based on the highest and minimum values in the training dataset, min–max scaling preserves the form of the dataset. The dataset is properly formatted and normalized thanks to these pretreatment steps, which make it suitable for training machine learning models.

The final step in this procedure is to ensure that the training dataset only contains normal data and the test dataset has both normal and anomalous data; the dataset must be divided into training, validation, and test sets. Partitioning is undertaken so that some features depend on how other features change over time to maintain the temporal order of the dataset. The training dataset contains the initial samples, while the test dataset contains the remaining samples. Furthermore, the validation dataset is derived from a tiny subset of the training dataset, usually in line with the 70/30 rule, but variations in the split size were also tested in some cases.

4.4. Algorithm Selection

The selection of appropriate algorithms for anomaly detection in industrial control systems (ICSs), such as those managing critical infrastructure, is a nuanced process that requires careful consideration of several factors. These include the nature of the dataset, the types of anomalies we aim to detect, and the operational constraints of the systems being monitored. For our project, which uses the SWaT July 2019 dataset, the criteria for selecting machine learning algorithms were grounded in the need for robustness, accuracy, real-time processing capabilities, and the ability to handle imbalanced data. The SWaT July 2019 dataset consists of sensor and actuator data from a water treatment testbed, and includes normal operational data and several documented attacks. The time-series data in the dataset exhibit a notable class imbalance, with a large number of normal operations compared to anomalies. These qualities dictated that our algorithm selection process had to concentrate on algorithms that are scalable for huge datasets, can handle class imbalance, and achieve remarkable success in time-series analysis.

4.4.1. Long Short-Term Memory

This algorithm is a type of recurrent neural network architecture that is intended to identify dependencies over time in sequential input. In contrast to conventional RNNs, which have trouble keeping information over lengthy periods due to the vanishing gradient problem, LSTM models have gated units called "memory cells" that control the information flow inside the network. Because of these memory cells, LSTM models are very good at modelling sequential data with long-range dependencies, as mentioned in the study by [34]. They can be applied for long periods of time to either forget or selectively recall information.

The research by [35] provides great insight into this algorithm. Three "gates" make up these memory cells: input, output, and a forget gate. LSTM networks can selectively retain or forget information over time thanks to these gates, which regulate the flow of information into and out of the cell.

Forget Gate: In an LSTM cell, as shown in Figure 1, choosing which data to discard from the cell state is the first step in handling a new input. The forget gate, which receives as inputs the current input and the previous concealed state, decides this. It produces a value between 0 and 1 for each cell state element. A zero indicates that it should "completely discard this info," and a 1 indicates that it should "wholly retain this information".

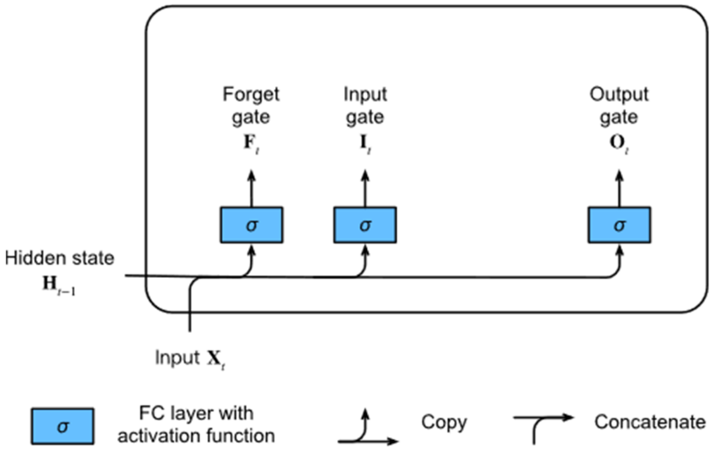
Forget gate $\mathbf{F}_t$
Input gate $\mathbf{I}_t$
Output gate $\mathbf{O}_t$
σ
σ
σ
Hidden state $\mathbf{H}_{t-1}$
Input $\mathbf{X}_t$
σ
FC layer with activation function
Copy
Concatenate

Figure 1. Long Short-Term Memory cell.

Input Gate: The input gate then decides what fresh data should be given to the cell state. It is composed of two layers: a tanh layer that generates a vector of new candidate inputs to add to the cell state, and a sigmoid layer that determines what data to update. Values between 0 and 1 are generated by the sigmoid layer, signifying the relative importance of each component in the candidate vector.

Cell State Update: The new candidate values are chosen by the input gate, and the information kept from the previous phase (decided by the forget gate) is combined with them to update the cell state. As a result, a new cell state is created that selectively keeps pertinent information while removing unnecessary information.

Output Gate: Lastly, the output gate chooses which data from the cell to output. It generates a filtered version of the cell state that is passed on to the following time step by taking as inputs the current input and the previous hidden state.

LSTM networks are especially helpful for applications like time-series prediction [36], as seen in natural language processing and speech recognition, because they can efficiently capture long-range dependencies in sequential data by employing these gates to govern the flow of information.

4.4.2. Random Forest

For classification tasks in machine learning, Random Forest is a widely used ensemble learning technique. It belongs to the family of decision tree-based algorithms and is renowned for being robust, flexible, and efficient when working with large, complicated datasets, as indicated in the study by [5]. The basic underlying idea in Random Forest is that during the training phase, multiple decision trees are generated and their predictions are aggregated to increase overall accuracy and generality.

The initial step in the process is to create multiple decision trees. Each tree is constructed using a subset of the training data and a random selection of features. Bagging, or bootstrap aggregating, is a technique for implementing this unpredictability. It is necessary to train each decision tree on a random subset of the training set when using replacement bagging. The study by [37] states that by training each tree on different data subsets, overfitting is minimized and the model's capacity for applicability to new, untested data is maximized.

Apart from deploying randomized subsets of the training data, Random Forest further increases unpredictability by selecting an arbitrary set of features at every decision tree split, as per [38]. As the trees are more diverse due to this feature randomization, the ensemble is more resilient and less prone to overfitting. Because of the combination of feature randomness and bagging, each tree in the forest is unique and represents a different portion of the data, strengthening the ensemble as a whole.

The prediction stage commences once the forest of decision trees is formed. For a given input sample, every decision tree independently predicts the class label. The final prediction is made by a majority voting technique, where the projected class is selected from among the trees based on which class received the most votes. Through the use of the collective insights from each individual tree, this voting technique yields more reliable and accurate projections.

Random Forest's ensemble learning approach improves the model's overall performance. Random Forest offsets the drawbacks of individual trees by blending their predictions, thus enhancing the model's generalization capabilities. Additionally, the randomness added during the training process provides inherent regularization, preventing overfitting and guaranteeing that the model performs well on both raw unseen data and training data. Therefore, Random Forest is an effective and extensively utilized method for classification applications. It is an appropriate option for ML/DL applications due to its capacity to handle high-dimensional data, manage missing values, and provide consistent performance across a range of datasets. The ensemble aspect of Random Forest, which consists of multiple decision trees and a voting mechanism, offers a dependable and efficient method to handle classification problems.

4.4.3. One-Dimensional CNNs

One-dimensional CNNs are a deep learning subset of the machine learning algorithm family that are designed to process sequential data, which makes them appropriate for use in critical infrastructure systems for tasks like anomaly detection and time-series analysis, as learned from [33]. In our case, 1D CNNs are made to handle one-dimensional data, such as the time series of sensor measurements, as opposed to standard CNNs, which work with 2D data, like photos. Convolutional, pooling, and fully linked layers are among the layers that make up a 1D CNN's architecture. To identify local patterns in the input data, the convolutional layers apply filters, also known as kernels, that move across the data. Every filter carries out a convolution operation, which entails taking a portion of the input data and the filter's dot product.

In the context of anomaly detection in critical infrastructure systems, the convolutional layers of a 1D CNN can learn to identify patterns associated with normal operating behavior and distinguish them from patterns indicative of anomalies, as per [33]. For example, a filter could be trained to detect abnormal increases in sensor readings that are indicative of certain kinds of attack. In order to minimize the dimensionality of the feature maps, save computing costs, and help make the features less sensitive to relatively minor variations in the input data, pooling layers are frequently used following convolutional layers. Standard pooling operations involve average pooling, which determines the average value, and max pooling, which chooses the highest value from each feature map segment.

The feature maps are typically flattened and fed into one (or more) fully connected layer after the convolutional and pooling layers. These layers use the collected features to undertake high-level reasoning. The output of the network's last layer is usually produced by a ReLu or sigmoid activation function. This function can yield a binary classification that indicates the existence or absence of an anomaly, or it can produce a probability distribution over several classes. The study by [39] outlines that 1D CNNs are very good at identifying anomalies in time-series data because they can recognize hierarchical patterns and local relationships in sequential data. They do not require human feature engineering because they can automatically extract pertinent characteristics from raw data. This capability is particularly beneficial in scenarios involving complex and high-dimensional data, such as the SWaT July 2019 dataset, where sensor readings and actuator states are monitored continuously.

4.4.4. Unsupervised Algorithms

We seek to evaluate two well-known unsupervised methods for anomaly detection in critical infrastructure systems (IDSs): isolation forests and the one-class support vector machine (SVM). We will compare their performance with supervised algorithms using the SWaT dataset.

The one-class SVM detects outliers by training a decision function on predominantly normal instances. It creates a boundary around the training data, usually generating a hypersphere or hyperplane. The objective is to maximize the gap between data points and the origin, with points beyond this limit classified as anomalies. As seen in [40], this method can discover small anomalies in complex datasets because it works well in high-dimensional spaces and can use the kernel trick to represent non-linear connections.

Isolation forests adopt a different strategy, specifically isolating anomalies using a collection of random decision trees. Each tree randomly chooses a feature and a split value, splitting the data until all points are isolated or a preset height is attained [18]. Because anomalies are infrequent and distinct, they are easier to isolate and have shorter path lengths. The average path length of every tree is used to compute the anomaly score. Isolation forests are efficient and scalable, making them ideal for handling big datasets found in critical infrastructure systems. While both isolation forests and Random Forests include numerous trees, their purposes are distinct. Isolation forests utilize random splits to isolate anomalies, determining outliers based on path lengths. In contrast, Random Forests integrate predictions from numerous trees to improve classification or regression accuracy. Isolation forests are straightforward and scalable, but they could overlook more complicated anomalies when compared to one-class SVMs, which require more resources but can handle complex data distributions.

4.5. Evaluation Metrics

In any research endeavor, evaluation metrics are essential for evaluating how well tools, algorithms, or procedures perform. We have included quantitative measures of accuracy, precision, recall, and F1 score in our evaluation. In order to comprehend the efficacy and effectiveness of the chosen machine learning algorithms, these indicators are essential. Some of the most relevant measurement metrices for machine learning algorithms are mentioned below. Which performance metric helps us find answers to our research inquisitions depends on the context of the study, along with the datasets used for the study. It is worth noting that not all metrics may offer a correct representation of how apt the proposed model is for anomaly detection.

Accuracy: The percentage of accurately categorized cases in the dataset is known as accuracy. The computation involves calculating the ratio of all cases with the amount of correctly identified instances (true positives and true negatives). Although accuracy offers a broad indicator of the performance of the model, imbalanced datasets where one class predominates over the other may not be appropriate for accuracy.

$$Accuracy = \frac{TP + TN + FP + FN}{(TP + TN)}$$

where

- TP = true positives (correctly classified attacks);
- TN = true negatives (correctly classified normal operations);
- FP = false positives (normal operations incorrectly classified as attacks);
- FN = false negatives (actual attacks incorrectly classified as normal operations).

Precision is a measure of the accurateness of a model's positive assumptions. It is defined as the ratio of true positive (*TP*) results to the total number of positive predictions made by the model, which includes both true positives and false positives (*FP*). In our context, precision tells us that out of all the instances that were detected as attacks or anomalies, how many of them actually were attacks on the system.

$$Precision = \frac{TP}{TP + FP}$$

Recall (also known as sensitivity or true-positive rate) is a metric that indicates how well a model can locate all pertinent occurrences in a dataset. It is defined as the ratio of true-positive results to the total number of actual positive instances, which includes both true positives and false negatives. In our context, recall tells us the number of attacks that were successfully detected from all the attacks that were carried out on the system.

$$Recall = \frac{TP}{TP + FN}$$

F1 score is a metric that provides a single assessment of the accuracy of a model by combining recall and precision. In situations where there is a discrepancy between classes, it is extremely beneficial, which is common in anomaly detection tasks. With a high F1 score, the model accurately identifies a large percentage of true positives while keeping a small quantity of false positives, demonstrating a strong balance between precision and recall. The following formula is used to compute it:

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

A Confusion matrix is a tabular representation of the counts of true-positive, true-negative, false-positive, and false-negative predictions that provides an overview of the performance of a classification algorithm.

4.6. Preferred Metric

In the context of our research, the F1 score is designated as the preferred metric of evaluation. This choice is justified by several critical factors. First, in anomaly detection, it is essential to not only identify true anomalies (high recall), but also to ensure that the anomalies detected are indeed correct (high precision). The F1 score offers a balanced metric that takes into account both factors because it is the harmonic mean of recall and precision. This balance is crucial in our scenario, where false positives can lead to unnecessary alarms and operational disruptions, while false negatives can allow security breaches to go undetected.

Second, the SWaT dataset, like many real-world datasets, exhibits a severe class imbalance, with far fewer anomalies compared to normal instances. In certain situations, metrics such as accuracy might be deceiving because a model may attain high accuracy by just forecasting the majority class. The F1 score, however, is more informative as it specifically focuses on the performance with respect to the minority class (anomalies), which makes it a better option to assess our models.

Third, the F1 score offers a solitary, comprehensive measure that captures the balance between recall and precision. This is particularly useful for comparing different models and hyperparameter settings in a clear and concise manner. It simplifies the assessment procedure by offering a comprehensive perspective of the model's functionality in detecting anomalies.

Finally, in the context of critical infrastructure systems, the consequences of both false positives and false negatives are significant. While false negatives can result in security breaches that go unnoticed and have serious repercussions, false positives can cause needless reactions and even shutdowns. The F1 score helps us find a model that optimally balances these risks, ensuring reliable and accurate anomaly detection. Given these considerations, the F1 score emerges as the preferred evaluation metric for the present study, which identifies the impact of machine learning on critical infrastructure systems using the SWaT July 2019 dataset.

By focusing on the balance between precision and recall, the F1 score ensures that our models are both accurate in identifying true anomalies and robust against generating false alarms. This makes it an ideal choice in the assessment of the usability of machine learning algorithms.

5. Experimental Analysis and Discussion

5.1. System Description

The Secure Water Treatment (SWaT) testbed is an actual-world environment designed for examining and experimentation within the domain of industrial control system (ICS) security. It serves as a simulation of a water treatment plant, mimicking the processes and infrastructure found in real-world critical infrastructure systems.

The SWaT testbed consists of various components that replicate the functionalities and processes of a typical water treatment plant. These components include the following:

Pumps and Valves:

The testbed comprises pumps and valves responsible for controlling the passage of water through various phases of the treatment process. These components are critical for regulating water pressure and ensuring the smooth operation of the system.

Tanks and Reservoirs:

Tanks and reservoirs store water at various stages of the treatment process. Their function is essential in preserving a steady flow of water and facilitating the different treatment procedures.

Sensors and Actuators:

Sensors are distributed throughout the testbed in order to gather information regarding various parameters like water flow, pressure, temperature, and chemical levels. Actuators are devices that respond to sensor data by controlling the operation of pumps, valves, and other components.

Control Systems:

The control systems in the SWaT testbed oversee the operation of the entire water treatment process. They take data from sensors, process the data using preset algorithms, and then instruct actuators to change the parameters of the system as necessary. The process testbed chart is shown in Figure 2.

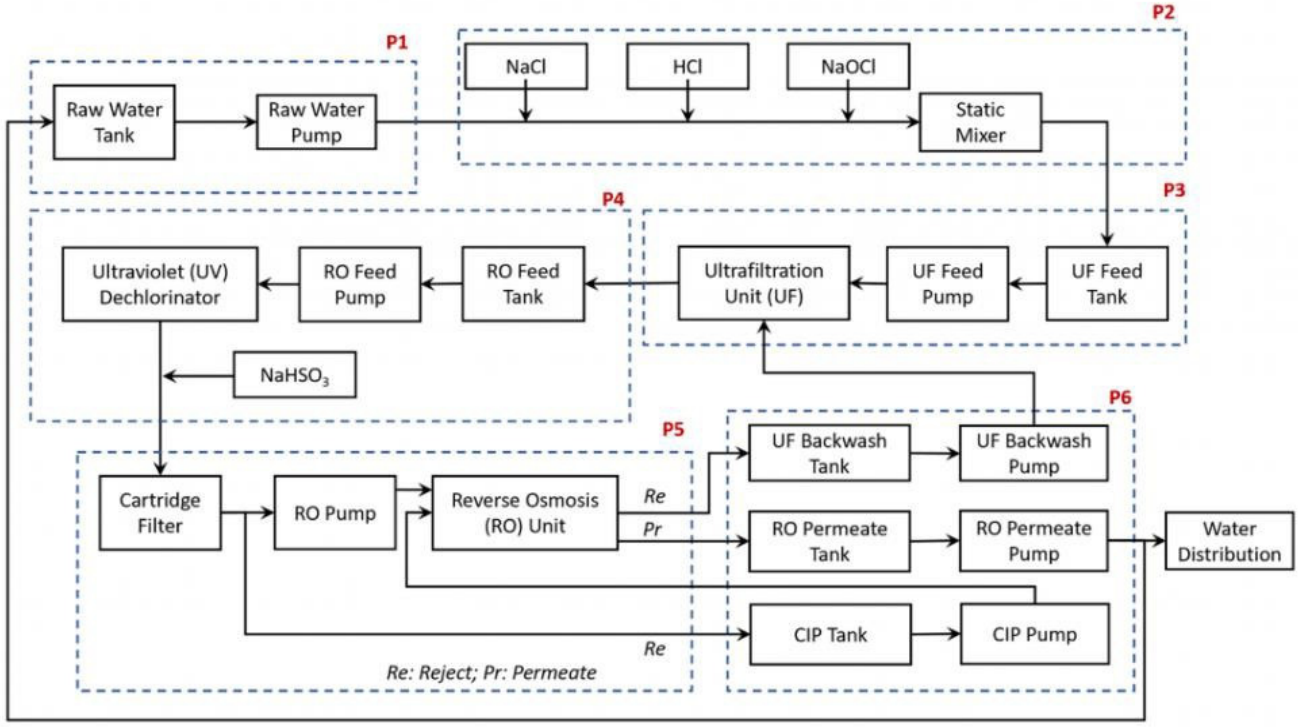

Figure 2. SWaT testbed process chart.

The Secure Water Treatment (SWaT) testbed is designed for emulating the functioning of a contemporary water treatment plant, comprising six distinct processes aimed at purifying water for potable use. Each stage in the SWaT testbed performs a vital part in the treatment process, guaranteeing the removal of impurities and the provision of clean, safe water to users.

Raw Water Intake (P1):

This stage controls the inflow of untreated water into the treatment system. A valve regulates the movement of water into the untreated water tank from the inflow pipe, where it awaits further processing.

Chemical Disinfection (P2):

In this stage, the raw water undergoes chemical treatment for disinfection purposes. Chlorination is performed using a chemical dosing station, which adds chlorine to the water to eliminate pathogens and bacteria.

Ultrafiltration (P3):

The water treated in the previous stages is sent to an Ultra-Filtration (UF) feed water storage unit. Here, a UF feed pump propels the water through UF membranes, removing suspended solids, microorganisms, and other impurities.

Dichlorination and Ultraviolet Treatment (P4):

Prior to passing through the reverse osmosis (RO) unit, the water undergoes dichlorination to remove any residual chlorine. Ultraviolet (UV) lamps are used for this purpose, controlled by a PLC. Additionally, sodium bisulphate may be added to regulate the Oxidation Reduction Potential (ORP) of the water.

Purification by Reverse Osmosis (P5):

After being dechlorinated, the water is filtered through three stages of reverse osmosis. RO membranes remove dissolved salts, organic compounds, and other contaminants, producing purified water that is reserved in an absorbent tank.

Ultrafiltration Membrane Backwash and Cleaning (P6):

This stage is responsible for maintaining the efficiency of the UF membranes by periodically cleaning them through a backwash process. A UF backwash pump initiates the cleaning cycle, which is triggered automatically based on differential pressure measurements across the UF unit.

Throughout these stages, various sensors and actuators collect data and control the operation of the equipment. PLCs (Programmable Logic Controllers) play a central role in orchestrating the treatment process, monitoring sensor readings, and executing control algorithms to ensure optimal performance and safety. Sensor data is logged and transmitted for real-time monitoring and analysis, facilitating the detection of anomalies or deviations from expected behavior.

5.2. Experimental Setup

The experiments for this research were conducted using a high-performance computing environment to manage the substantial computational demands of training and evaluating machine learning models on the SWaT dataset. The setup featured an ASUSTek AMD Ryzen 7 5800H (R) CPU @ 2.20 GHz with 16 virtual CPUs, 16 GB of system memory, and an NVIDIA GeForce GTX 1650 GPU with 4 GB of VRAM (manufactured in Mumbai, India). The software environment comprised Python 3.10.12 as the primary programming language, with TensorFlow 2.13.0 for deep learning models. Traditional machine learning models were implemented and tuned using Scikit-learn. Additionally, NumPy 1.24.4 and Pandas 2.1.1 were employed for data manipulation and preprocessing, while Matplotlib 3.7.2 and Seaborn 0.12.2 were utilized for visualizing results and performance metrics. The code was compiled and executed on Microsoft Visual Studio Code IDE 1.88.0. This robust experimental setup ensured that the models were trained and evaluated effectively, providing reliable insights into the application of machine learning for anomaly detection in critical infrastructure.

5.3. Attack Profile

Table 2 includes the details for the SWaT testbed, and some key characteristics are listed below:

- Plant operation time: 12:35:00 to 16:39:00.
- Plant start time: 12:35:00 (GMT +8).
- Normal run without any attacks: 12:35:00 to 15:06:59.
- Attack period: 15:07:00 to 16:15:07.
- Plant stop time: 16:39:00.

Table 2. SWaT testbed attack profile.

Attack	Target	Action	Intent	Start Time	End Time
1	FIT401	Spoof value from 0.8 to 0.5	To stop de-chlorination by switching off UV401	15:07:00	15:08:44
2	LIT301	Spoof value from 835 to1024	To eventually lead to underflow in T301	15:13:27	15:17:47
3	P601	Switch from OFF to ON	To increase water in raw water tank	15:25:13	15:29:02
4	Multi-point	Switch from CLOSE to OPEN (MV201) and OFF to ON (P101)	To overflow tank T301	15:37:12-MV201/15:37:19-P101	15:44:55-MV201/15:44:48-P101
5	MV501	Switch from OPEN to CLOSE	To drain water from RO	15:52:25	15:54:48
6	P301	Switch from ON to OFF	To halt stage 3 (UF process)	16:01:06	16:15:07

5.4. Data Pre-Processing of the SWaT Dataset

Upon receipt of the SWaT dataset in XLSX format, it underwent several preprocessing steps to ensure its suitability for analysis. Initially, the dataset was converted to CSV format for ease of handling. Subsequently, rigorous data cleaning procedures were implemented, including the removal of extraneous markings and headers. Quality checks were conducted to identify and address missing values (NaN) and outliers.

Normalization techniques were applied to features exhibiting high variance, ensuring uniformity and comparability across the dataset. Furthermore, features with discrete states were encoded using techniques such as one-hot encoding and label encoding to facilitate compatibility with machine learning algorithms.

The removal of sensor components and actuator values that remained constant throughout the dataset was carefully considered to reduce the possibility of overfitting of the machine leaning models. These unchanging values could potentially bias machine learning models, leading to inaccurate results.

Additionally, timestamp data within the dataset was standardized to a consistent datetime format, enabling easier manipulation and analysis. To facilitate supervised learning tasks, an "Attack" column was added to the dataset, labelling data corresponding to periods of simulated attacks as provided within the dataset documentation. The addition of the attack column was necessary for classification in the algorithms used for anomaly detection. This was performed based on the timestamp data of attack initiation and termination provided by iTrust along with the dataset. After adding the attack labels, the volume of normal and attack classes was checked. It was discovered that there was a severe class imbalance in the dataset when it came to normal and attack classes. In machine learning, class imbalance is a prevalent problem [40], predominantly in scenarios where one class ("normal" instances) heavily outweighs the other class ("attack" instances). Addressing class imbalance in machine learning datasets, particularly in scenarios where one class significantly outweighs the other, is crucial to prevent biased models and ensure accurate predictions. As per the research by [41], several methods are commonly employed to mitigate class imbalance before utilizing machine learning algorithms for classification. Resampling techniques try to balance the class distribution by either replicating instances from the minority class, deleting instances from the majority class, oversampling the minority class, or under sampling the majority class. Class imbalance can be addressed by algorithms, such as ensemble techniques like Random Forest and Gradient Boosting, which give misclassified instances of the minority class more significance during training. Assigning varying costs to misclassification errors for every class is a component of cost-sensitive training, encouraging models to focus on correctly predicting the minority class instances, as mentioned by [42]. Additionally, techniques for detecting anomalies handle the minority class as anomalies or outliers, identifying deviations from the majority class as anomalies. Figure 3 shows the distribution of classes in the dataset.

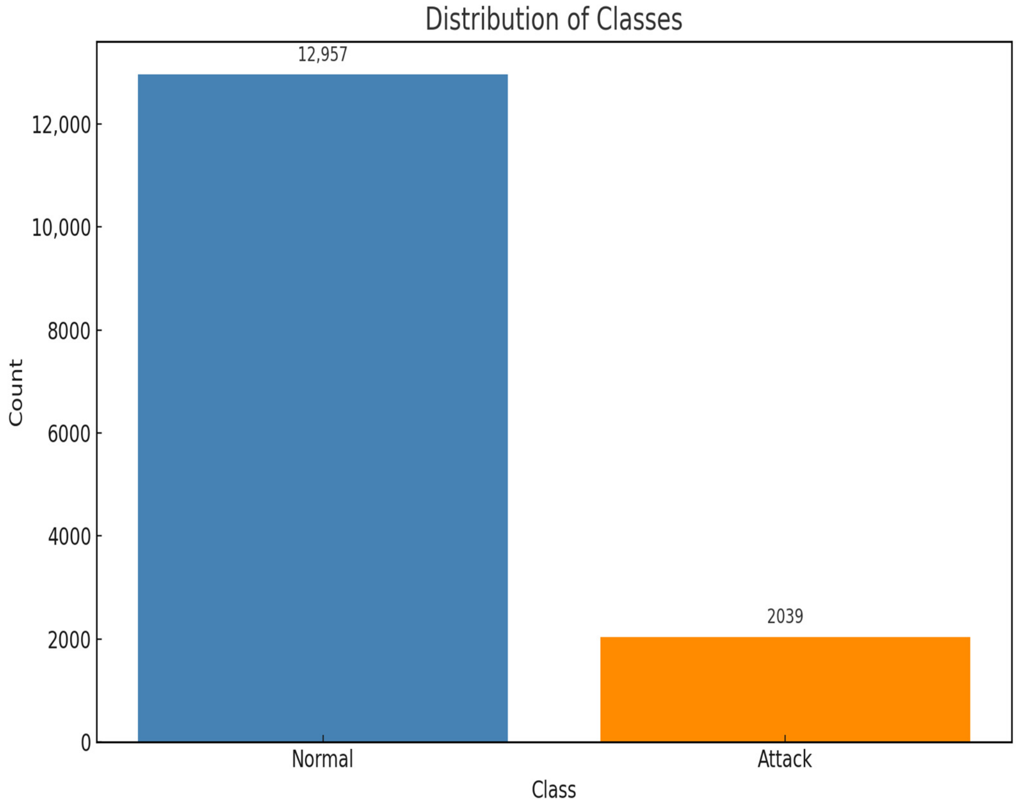

Figure 3. Normal and attack instances in the dataset.

Ref. [43] states that while the methods mentioned above can help mitigate class imbalance, they may not always be suitable for real-world critical infrastructure datasets due to concerns about data integrity, impact on performance, and the cost of misclassification. Generating synthetic data or undersampling the majority class can compromise data integrity, while oversampling techniques may lead to overfitting and poor generalization performance. Moreover, this sentiment is reinforced in the study by [44], which outlines that the misclassification of attacks as normal behavior (false negatives) can have severe consequences in critical infrastructure, necessitating careful consideration of the costs associated with misclassification errors. Therefore, when working with real-world critical infrastructure datasets, it is essential to evaluate the implications of addressing class imbalance and choose appropriate methods that prioritize model robustness, reliability, and interpretability. Expert input and domain knowledge play a crucial role in making informed decisions about how to handle class imbalance effectively while mitigating the risks associated with misclassification. Owing to these reasons, it was decided to not address the class imbalance to achieve results as close to real-world scenarios as possible.

5.5. Feature Selection

Careful consideration of feature selection for the SWaT July 2019 dataset is crucial due to its high dimensionality and complexity, severe class imbalance, and the operational significance of each feature. In critical infrastructure systems like water treatment processes, selecting the most relevant features reduces noise and redundancy, enhancing the capacity of the model to discern between normal and anomalous behaviors. This leads to improved detection accuracy and model performance, which is essential for identifying subtle indicators of anomalies or attacks. Effective feature selection also improves model interpretability, efficiency, and scalability, making the models faster to train and deploy. Ultimately, this process ensures the safety and reliability of the infrastructure by providing robust and accurate anomaly detection, which is vital for protecting against potential cyber threats and operational failures.

In the context of the SWaT July 2019 dataset, which involves monitoring a water treatment facility, constant features provide no informational value regarding the state or behavior of the system. These features do not contribute to the detection of anomalies or attacks because they lack variability and hence cannot indicate any deviation from the norm. Including constant features would add unnecessary complexity to the model without improving its performance, corroborated by the study by [28], as machine learning and deep learning models rely on variability amongst data to learn patterns and differentiate between normal and anomalous states. Additionally, the presence of constant features can lead to overfitting, where the model learns noise and irrelevant details, resulting in inadequate generalization to novel, unforeseen data.

Removing constant features enhances computational efficiency by reducing the computational load of training the model, making the process faster and less resource intensive. Simplifying the feature set also improves interpretability, which is critical in industrial control systems (ICS) like SWaT, where understanding the model's decisions is essential for identifying possible security issues and taking appropriate action. Thus, in coherence with the research by [29], dropping constant features is an essential preprocessing step that improves model accuracy, prevents overfitting, enhances computational efficiency, and maintains interpretability, all of which are crucial for effective anomaly detection in critical infrastructure systems. In contrast, highly correlated features were selectively retained if they reflected redundant but critical process variables—for example, flow rate and corresponding valve state. During feature selection, we retained sensor and actuator readings that demonstrated temporal variability, operational relevance, or a historical association with known attack vectors in the SWaT documentation. This decision was based on domain knowledge indicating that correlated features in industrial control systems can still provide independent forensic value when an attacker manipulates only one aspect of the process. Moreover, retaining key actuator states enabled the models to better identify coordinated attacks that modified multiple components simultaneously.

Next, we moved on to analyzing the correlations between different features of this dataset. Understanding these correlations allows us to detect anomalies more effectively by recognizing deviations from established patterns. However, we decided to not drop certain highly correlated features, even if they seem redundant, because they represent critical sensors and actuators integral to the water treatment process. These elements provide essential information about the system's state and operations, and their interactions can significantly impact the detection of both normal and malicious activities. Retaining these features ensures that the IDS has a comprehensive view of the system, enabling more accurate and reliable anomaly detection.

5.6. Machine Learning Implementation and Discussion

A one-dimensional CNN is suitable for the detection of anomalies in critical infrastructure owing to its capability to capture local patterns and dependencies in sequential data. By applying convolutional filters along the temporal dimension, as stated in [45], 1D CNNs can identify subtle deviations or irregularities indicative of security breaches or system anomalies. Additionally, 1D CNNs can capture intricate temporal patterns because they automatically learn structured representations of the input data and the relationships present in critical infrastructure data, further enhancing their anomaly detection capabilities.

We began the analysis by strategically exploring the best configuration for the model to achieve the best performance with the minimum cost to the system.

In our study, we experimented with various hyperparameters, shown in Table 3, to optimize the performance of the 1D CNN. The selected configuration (parameters in bold) included the Adam optimizer, 64 filters, a kernel size of 7, a dropout rate of 0.2, 25 epochs, and a batch size of 32. These hyperparameters were chosen based on their ability to balance training speed and model accuracy while preventing overfitting. Figure 4 shows the training and validation loss for the 1D CNN, and Figure 5 shows its training and validation accuracy.

Table 3. Hyperparameter sets for 1D CNN.

Hyperparameters	**Values**
Optimizer	**Adam**, rmsprop
Filters	32, **64**
Kernel size	3, 5, **7**
Dropout rate	**0.2**, 0.3, 0.4
Epochs	10, 15, 20, **25**
Batch size	**32**, 64

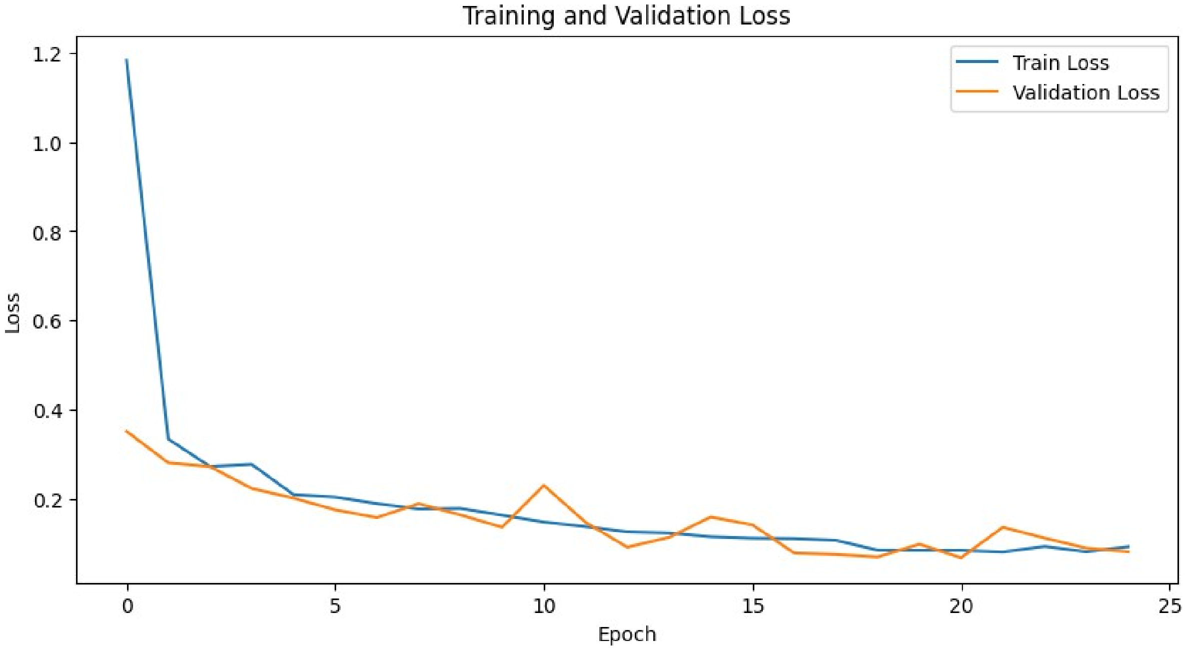

Figure 4. Training and validation loss for 1D CNN.

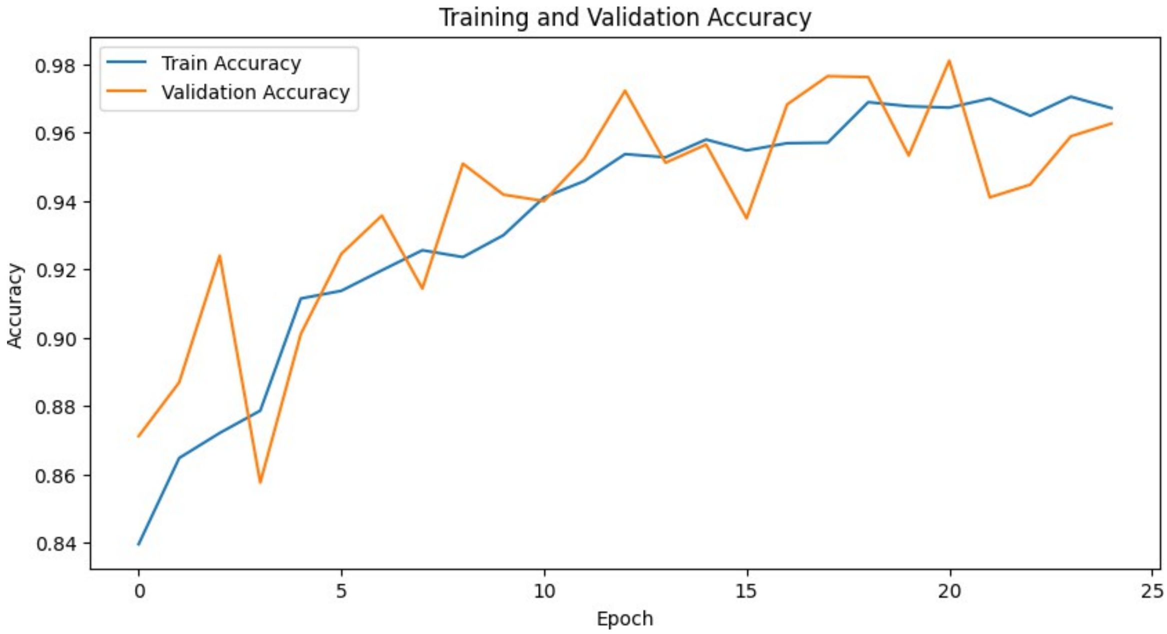

Figure 5. Training and validation accuracy for 1D CNN.

One of the key strengths of the 1D CNN is its ability to capture temporal patterns and dependencies in data. This is particularly important for the SWaT dataset, where the sequence of events and time-stamped sensor readings play a critical role in identifying anomalies. By using convolutional layers, the 1D CNN can automatically learn and extract relevant features from the raw data, reducing the need for extensive manual feature engineering.

Table 4 lists the performance results for 1D CNN. The accuracy of 96.71% indicates that the 1D CNN can reliably distinguish between normal and anomalous events, making it a valuable tool for monitoring and protecting critical systems—the SWaT testbed in this instance. The precision of 87.35% reflects the model's effectiveness in correctly identifying true-positive anomalies, reducing the occurrence of false positives, which is essential for maintaining operational efficiency and avoiding unnecessary disruptions. The recall of 89.81% shows that the model can detect a high proportion of actual anomalies, ensuring that potential security threats are identified promptly. The F1 score of 88.13% balances precision and recall, providing a comprehensive measure of the model's overall performance. The Confusion matrix for the 1D CNN is shown in Figure 6.

Table 4. Performance of 1D CNN.

True Positives (TP): 448	Accuracy: 0.9671
False Positives (FP): 66	Precision: 0.8735
True Negatives (TN): 3178	Recall: 0.8981
False Negatives (FN): 57	F1 Score: 0.8813

However, the 1D CNN also has some limitations. Training deep learning models like CNNs can be computationally intensive and time-consuming, requiring significant processing power and memory resources. This can pose challenges in real-time monitoring scenarios where rapid detection and response are crucial. Additionally, deep learning models can be difficult to interpret, making it challenging to understand and explain their decision-making processes. This lack of transparency can be a drawback in critical infrastructure applications, where accountability and explainability are important for ensuring trust and compliance.

Despite these challenges, the 1D CNN's ability to handle large volumes of sequential data and its strong performance metrics make it a promising candidate for anomaly detection in critical infrastructure. Its ability to learn and adapt to complex patterns in the data provides a significant advantage over traditional methods, which may struggle to capture the same level of detail and nuance.

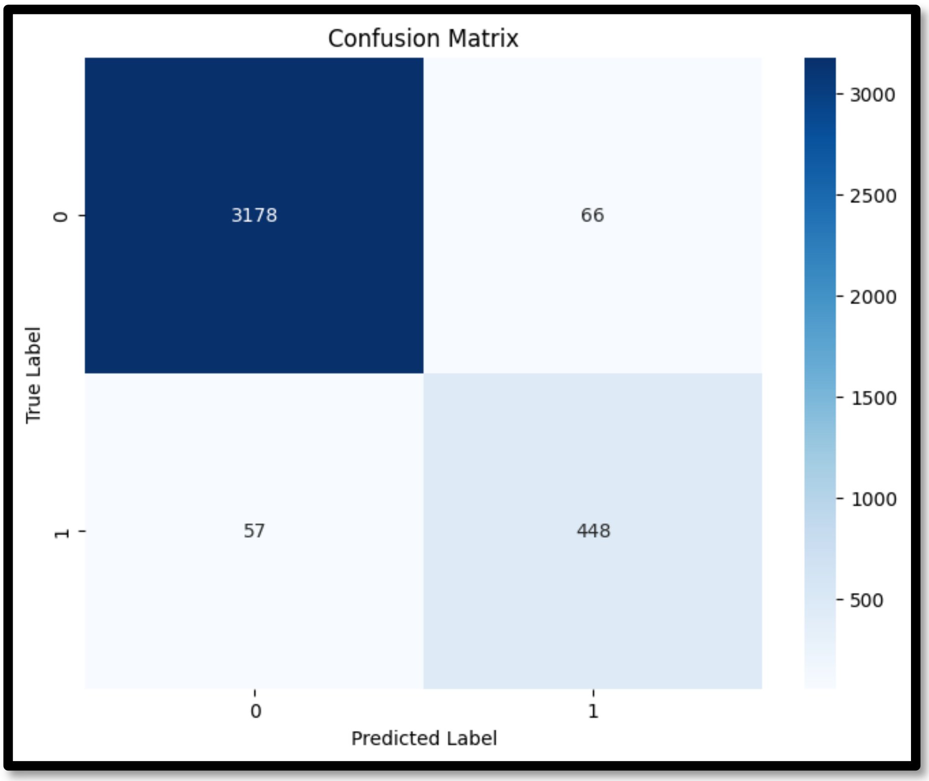

Figure 6. Confusion matrix for 1D CNN.

In the context of existing research, our findings align with studies that have demonstrated the effectiveness of CNNs for anomaly detection in time-series data. For example, Ref. [21] emphasized the utility of CNN-based models in detecting anomalies in industrial control systems, showing their potential in improving security and reliability. Similarly, Ref. [45] highlighted the advantages of using CNNs in critical infrastructure settings, noting their ability to process large-scale data efficiently and accurately. Furthermore, the work by [44] demonstrated that CNNs could effectively detect cyberattacks in smart grid environments, reinforcing the model's applicability to various critical infrastructure domains.

Overall, the 1D CNN offers a robust solution for detecting anomalies in critical infrastructure systems. Its ability to process and analyze sequential data, combined with its strong performance metrics, makes it a valuable tool for enhancing the security and resilience of these systems. However, the computational demands and interpretability challenges associated with deep learning models must be carefully managed to ensure their successful deployment in real-world applications.

5.6.1. Long Short-Term Memory (LSTM)

For our LSTM model, we chose to keep the feature selection and setup as close to the real-world scenario as possible. We initially chose even the static features to test the model, but that resulted in model overfitting. So as discussed in the research methodology, we chose to remove the static features and proceeded to build and train our model. We tested multiple sets of hyperparameters for the LSTM, and they are listed in Table 5 with the selected parameters in bold. Unlike CNNs, which are typically used for spatial data, LSTM models are designed to capture temporal dependencies by maintaining a memory of previous inputs. This capability is crucial for identifying anomalies in datasets where the order and timing of events are important, such as the SWaT dataset.

Table 5. Hyperparameter sets for LSTM.

Hyperparameters	**Values**
Units	32, 64, **128**, 256
Sequence length	1, **5**, 10, 50
Batch size	32, **64**, 128
Learning rate	0.1, 0.01, **0.001**
Epoch	10, 15, **20**, 25
Optimizer	**Adam**, RMSprop, SGD
Dense	**1**, 2

In this study, the LSTM model was configured with 128 units, a sequence length of 5, a batch size of 64, a learning rate of 0.001, 20 epochs, the Adam optimizer, and a single dense layer. These hyperparameters were selected to balance the complexity and performance of the model, ensuring it could effectively learn from the time-series data without overfitting. The training and validation loss for LSTM is shown in Figure 7.

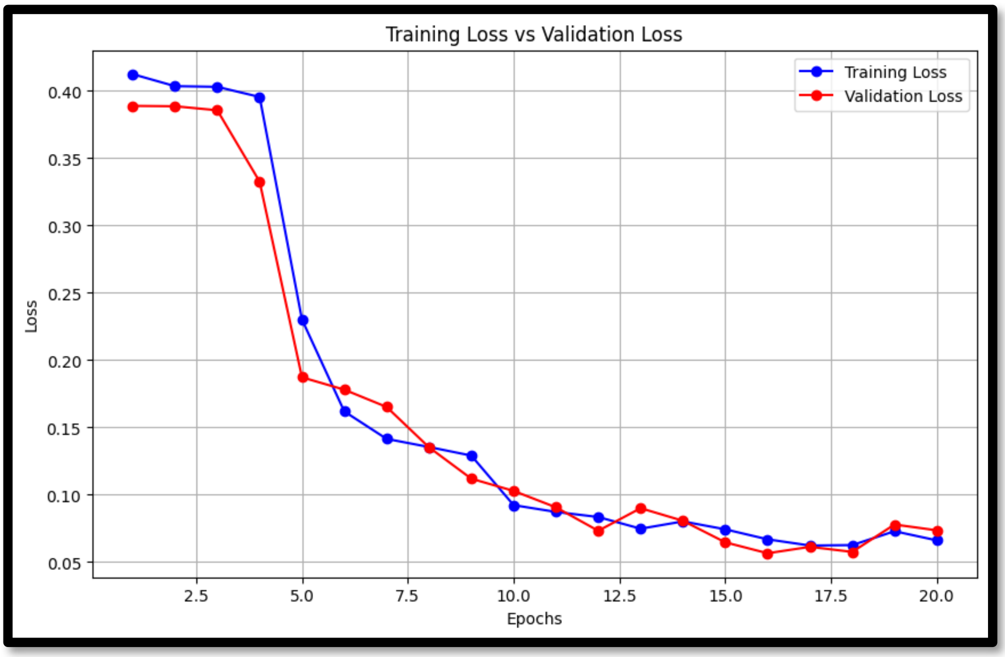

Figure 7. Training and validation loss for LSTM.

The following are the results from the optimal hyperparameter set of the LSTM model (Table 6):

Table 6. Performance of LSTM.

True Positives (TP): 481	Accuracy: 0.9722
False Positives (FP): 18	Precision: 0.9639
True Negatives (TN): 3893	Recall (Sensitivity): 0.8180
False Negatives (FN): 107	F1 Score: 0.8850

The high accuracy (97.22%) indicates LSTM's proficiency in distinguishing between normal operation and attacks, which is critical for effective intrusion detection. Precision (96.39%) demonstrates the model's ability to correctly identify true positives, minimizing false alarms and ensuring that security resources are allocated efficiently. The recall (81.80%) reflects the model's capability to capture a significant proportion of actual anomalies, although there is still potential for improvement in detecting all threats. The F1 score (88.50%) provides a balanced measure of overall performance, incorporating both precision and recall. These results highlight LSTM's ability to accurately detect anomalies in the SWaT dataset, which is essential for maintaining the security and operational integrity of critical infrastructure.

The primary advantage of LSTM is its ability to handle long-term dependencies and capture temporal patterns in data. This comes into effect on the SWaT dataset, where the sequence and timing of sensor readings are crucial for identifying anomalies. The memory cells in LSTM models allow them to retain important information over extended sequences, which provides a significant advantage over traditional anomaly detection methods that may not effectively capture such temporal dependencies.

However, LSTM models also present some challenges: the complex nature of LSTM models makes them difficult to interpret, posing challenges in understanding and explaining their decision-making processes. This lack of transparency can be a drawback in critical infrastructure applications, where accountability and explainability are essential for ensuring trust and compliance.

Despite these challenges, LSTM's strong performance metrics and ability to handle sequential data make it a promising tool for anomaly detection in critical infrastructure. Its ability to learn and adapt to complex patterns in the data provides a significant advantage over traditional methods, which may struggle to capture the same level of detail and nuance.

In comparison to the 1D CNN, which focuses on spatial patterns, the LSTM's strength lies in its ability to model temporal sequences. This makes it particularly effective for time-series data, where understanding the order of events is crucial. While the 1D CNN showed strong performance in anomaly detection, LSTM's ability to remember and utilize past information offers a complementary approach that enhances its overall detection capability. The Confusion matrix for LSTM is shown in Figure 8.

Our findings align with existing research highlighting the effectiveness of LSTM models for anomaly detection in time-series data. For example, Ref. [46] demonstrated the superiority of LSTM models over traditional models for sequence prediction and anomaly detection in industrial systems, showcasing their potential for improving security and reliability [46]. Similarly, Ref. [27] emphasized the advantages of using LSTM models for detecting cyberattacks in industrial control systems, noting their ability to accurately model complex temporal dependencies. Ref. [47] also reinforced the applicability of LSTM models to various critical infrastructure domains, highlighting their efficiency in processing large-scale time-series data.

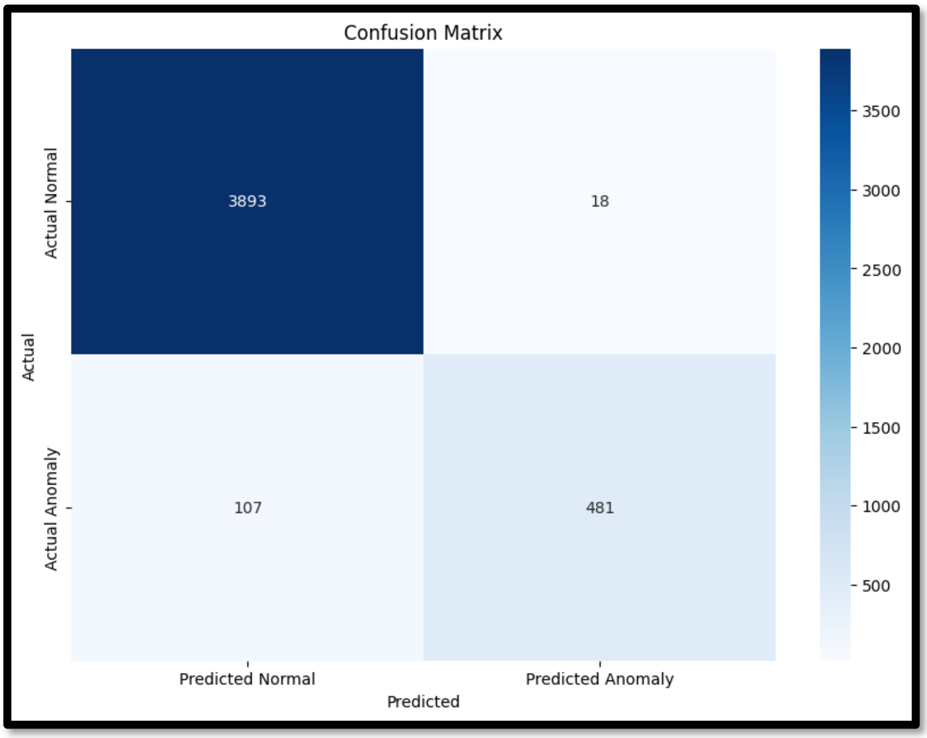

Figure 8. Confusion Matrix for LSTM.

It can thus be stated that LSTM offers a robust solution for detecting anomalies in critical infrastructure systems. Its ability to process and analyze sequential data, combined with its strong performance metrics, makes it a valuable tool for enhancing the security and resilience of these systems. However, the computational demands and interpretability challenges associated with deep learning models must be carefully managed to ensure their successful deployment in real-world applications.

5.6.2. Random Forest

Random Forest is suitable for anomaly detection in critical infrastructure systems due to its ensemble learning approach, which combines multiple decision trees to improve overall prediction accuracy and robustness. This allows Random Forest to capture complex relationships and patterns present in critical infrastructure data, enhancing its ability to detect anomalies, as seen in the earlier study by [13]. Additionally, Random Forest is a non-parametric learning algorithm, meaning it makes no assumptions about the underlying distribution of the data, making it flexible and adaptable to diverse data distributions commonly encountered in critical infrastructure systems. Table 7 lists the hyperparameter sets for Random Forest with selected parameters in bold, and its performance can be seen in Table 8.

Table 7. Hyperparameter sets for Random Forest.

Hyperparameters	**Values**
Trees	10, **25**, 50, 100
Features	auto, **sqrt**, log2
Max depth	**None**, 10, 20, 30, 40, 50
Minimum sample split	**2**, 5, 10
Minimum leaf sample	**1**, 2, 4
Bootstrap	**True**, False

Table 8. Performance of Random Forest.

True Positives (TP): 993	Accuracy: 0.9989
False Positives (FP): 7	Precision: 0.9930
True Negatives (TN): 6497	Recall: 0.9990
False Negatives (FN): 1	F1 Score: 0.9960

In our experiments, we evaluated Random Forest with a range of hyperparameters. The chosen configuration included 25 trees, a feature subset size of the square root of the total number of features, no maximum depth constraint, a minimum sample split of 2, a minimum leaf sample of 1, and bootstrap sampling enabled. This configuration was selected after rigorous hyperparameter tuning to optimize performance.

While splitting the dataset for Random Forest, we decided to test various combinations of data splits. We began with the usual 70/30 ratio, which was observed to produce good results. Since Random Forests are known for their resilience to overfitting, we decided to train the model on half of the dataset and test and validate the model's performance on the remaining half of the dataset. We decided to allocate 50 percent of the test data towards validation. This resulted in robust training of the model, albeit with a lower number of attack instances, which further tested the efficiency of the algorithm. The similarity in training vs. testing accuracy confirmed that the model was not prone to overfitting and performed well in detecting both normal and attack classes.

The performance metrics for Random Forest were impressive, with an accuracy of 99.89%, a precision of 99.30%, a recall of 99.90%, and an F1 score of 99.60%. These results demonstrate the algorithm's exceptional ability to accurately classify normal and anomalous events in the SWaT dataset. The high precision indicates that the Random Forest model is highly effective in correctly identifying true positives, minimizing the rate of false alarms, which is crucial for operational environments where false positives can lead to unnecessary interventions and increased operational costs. The near-perfect recall ensures that almost all actual anomalies are detected, highlighting the model's reliability in identifying security threats. The Confusion matrix for Random Forest is shown in Figure 9.

Random Forest's robustness to overfitting, due to its ensemble approach of combining multiple decision trees, contributes to its high performance. This characteristic is particularly beneficial in the context of critical infrastructure, where data can be noisy and include various operational conditions, as per [38]. In our context, this level of performance, although encouraging, could also be because of localized factors such as the small size of attack classes in the dataset, as well as slight variations in the readings during the attack. Since the variations are extremely small, and the model is trained on the normal data, it becomes highly sensitive to slight variations in the data pattern. Also, half of the test data was dedicated towards validating the training process, which further improved the model's ability to clearly distinguish between false positives and false negatives. The algorithm's ability to handle many features without significant loss of accuracy makes it well-suited to the complex and multifaceted nature of industrial control systems. In the context of critical infrastructure, the application of Random Forest for IDSs provides significant advantages. Its high detection accuracy ensures that most intrusions can be identified promptly, thereby mitigating potential damage to critical systems. Figure 10 shows the performance metric variation by epochs for Random Forest.

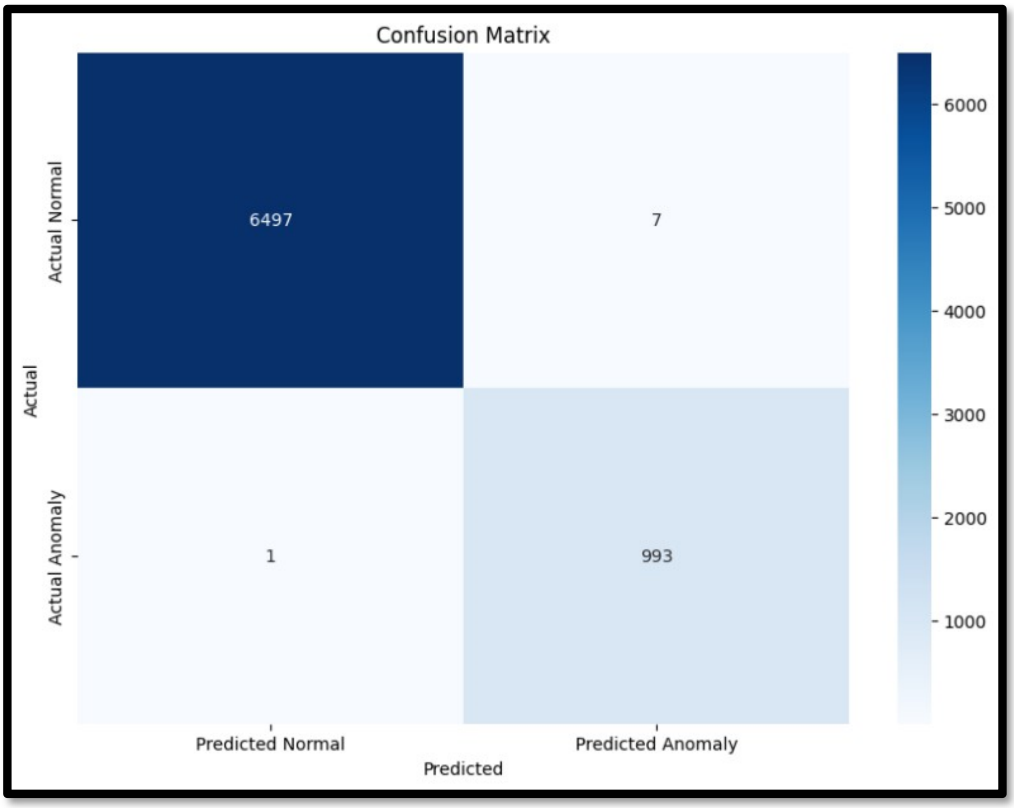

Figure 9. Confusion matrix for Random Forest.

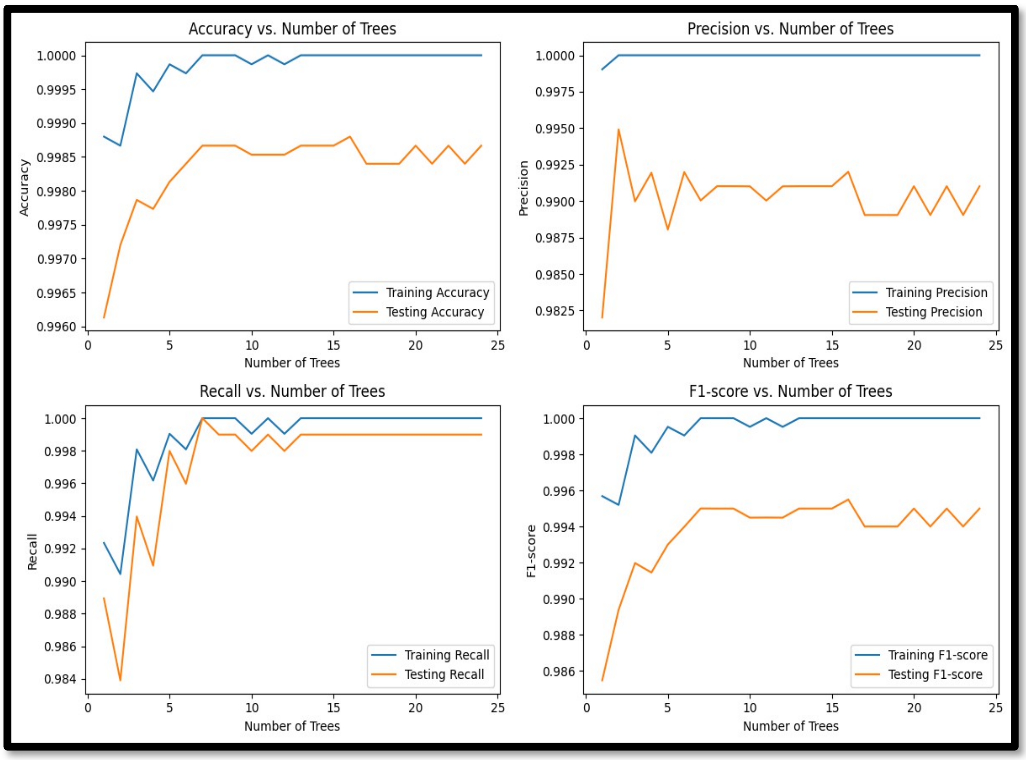

Figure 10. Performance metric variation by epochs for Random Forest.

Despite these strengths, the Random Forest algorithm has a major drawback. The interpretability of Random Forest models can be challenging. While the ensemble approach enhances predictive performance, it also makes it harder to understand the decision-making process of the model, which is a crucial aspect in security applications where transparency and accountability are important.

Comparing our findings with existing research, Random Forest has consistently shown strong performance in anomaly detection within industrial control systems. Studies such as those by [48] have also reported high detection rates and low false-positive rates, corroborating our results. This consistency across different studies and datasets underscores the reliability of Random Forest as a robust tool for enhancing the security of critical infrastructure. It can clearly be seen that the application of Random Forest in IDSs for critical infrastructure showcases its capability to accurately and reliably detect anomalies. The model's strengths in handling high-dimensional data and its robustness against overfitting make it an asset in the cybersecurity toolkit for protecting critical systems. However, attention must be paid to its need for interpretability in security applications.

5.6.3. One-Class Support Vector Machine (One-Class SVM)

The one-class support vector machine (one-class SVM) is an alternative form of the traditional support vector machine (SVM) algorithm intended for novelty recognition or outlier detection tasks. Unlike the traditional SVM, which is primarily used for binary classification, the one-class SVM learns a decision boundary that encompasses most of the information in a particular class ("normal" class), thereby identifying outliers or anomalies that deviate from the normal class.

For the SWaT dataset, the one-class SVM was configured with the linear kernel, a nu parameter of 0.1, and the gamma parameter set to "scale" (refer to Table 9 with the selected parameters in bold). The choice of the linear kernel helps in maintaining simplicity and interpretability, while the nu parameter defines an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors, controlling the trade-off between the training error and model complexity. The "scale" setting for gamma is based on the inverse of the number of features, ensuring that each feature contributes equally to the decision boundary.

Table 9. Hyperparameter set for one-class SVM.

Hyperparameters	**Values**
Kernel	**linear**, poly, rbf, sigmoid
Nu	**0.1**, 0.5, 0.7, 0.9
Gamma	**Scale**, auto

The one-class SVM is suitable for anomaly detection in critical infrastructure systems due to its novelty detection capabilities [23]. It learns the normal behavior of the system and detects deviations or abnormalities indicative of security breaches or system malfunctions. The one-class SVM is inherently robust to imbalanced datasets according to [24], where anomalies are relatively rare compared to normal system behavior, making it suitable for anomaly detection in critical infrastructure systems. Additionally, the one-class SVM utilizes a kernel function to map input data into a high-dimensional feature space, allowing it to capture complex relationships and non-linear decision boundaries present in critical infrastructure data, further enhancing its anomaly detection capabilities.

The performance metrics of the one-class SVM on the SWaT dataset reveal an accuracy of 85.09%, indicating a solid performance in distinguishing between normal and anomalous events. The precision of 0.4785 and recall of 0.5944 show that while the model is relatively balanced in identifying anomalies, it has a notable number of false positives and false negatives. The F1 score of 0.5261 further illustrates this balance between precision and recall. The ROC area of 0.74 signifies that the model's ability to distinguish between the two classes is fairly good, but there is room for improvement. Table 10 lists all the performance metrics for the one-class SVM.

Table 10. Performance of one-class SVM.

True Positives (TP): 363	Accuracy: 0.8509
False Positives (FP): 412	Precision: 0.4785
True Negatives (TN): 3475	Recall (Sensitivity): 0.5944
False Negatives (FN): 249	F1 Score: 0.5261

In the context of the SWaT dataset, the one-class SVM's performance can be explained by the inherent complexity and variability of the sensor data. The dataset includes numerous sensor readings under normal conditions and several simulated attack scenarios. Anomalies in this context are often subtle and context-dependent, such as slight deviations in sensor readings that indicate malicious activity. The linear kernel's simplicity might struggle with capturing these subtle and complex patterns, which could explain the number of false positives and negatives observed.

Despite these challenges, the one-class SVM remains a valuable tool for anomaly detection in critical infrastructure. Its effectiveness is demonstrated in scenarios where the normal operational patterns are well-defined and the anomalies are distinctly different, as in [49]. Figure 11 shows the Confusion matrix for the one-class SVM.

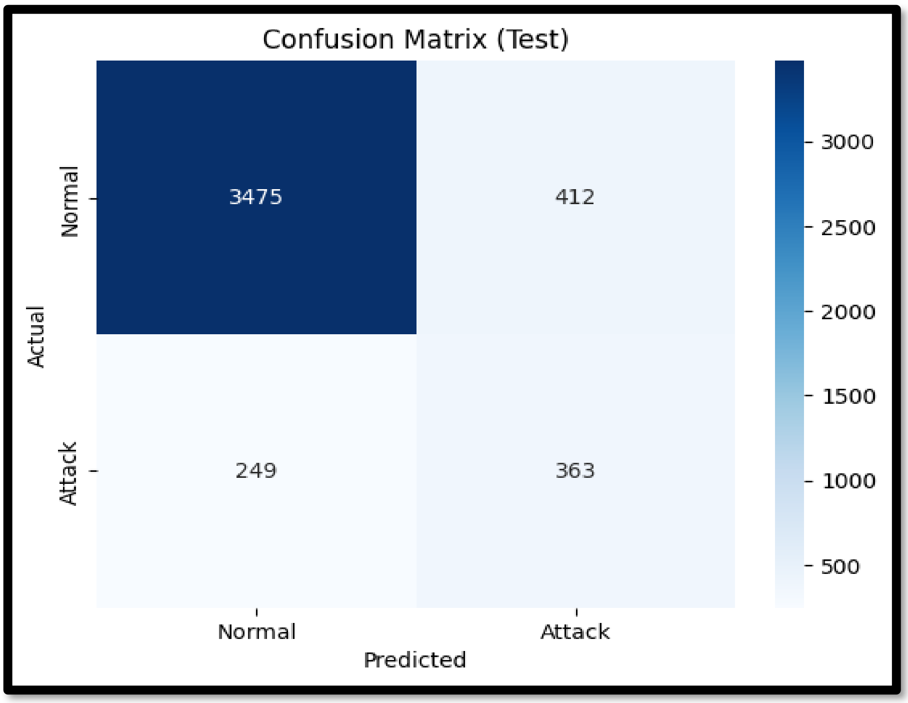

Figure 11. Confusion matrix for one-class SVM.

The relatively high ROC area (Figure 12) indicates that the model can separate normal from abnormal events better than random chance, which is crucial for the timely detection and mitigation of attacks on critical infrastructure.

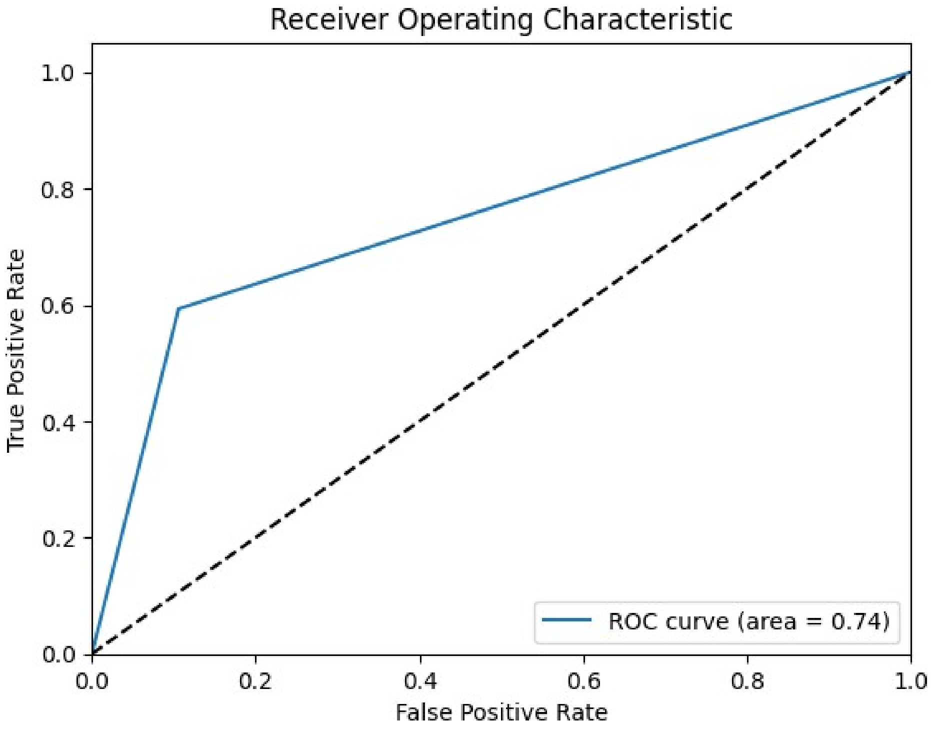

Figure 12. ROC characteristics for one-class SVM.

Improving the performance of the one-class SVM in this domain could involve experimenting with different kernels, such as the radial basis function (RBF), which might capture the non-linear relationships in the data more effectively. Adjusting the nu parameter to a higher value could also reduce the number of false negatives, as it would allow the model to consider more support vectors, thereby capturing more complex patterns. Additionally, feature engineering and normalization can enhance the model's ability to discern anomalies by ensuring that all features contribute meaningfully to the decision boundary. Integrating the one-class SVM with other anomaly detection techniques, such as supervised learning models trained on a subset of labeled data, can further improve its performance. This hybrid approach can leverage the strengths of unsupervised detection while incorporating the precision of supervised models, thereby providing a more comprehensive and reliable detection mechanism.

In conclusion, the one-class SVM offers a robust method for unsupervised anomaly detection in critical infrastructure. Its application to the SWaT dataset highlights its potential and the challenges inherent in detecting subtle and context-dependent anomalies. The model's performance underscores the need for continuous refinement and integration with other detection techniques to enhance the resilience of critical infrastructure systems against evolving cyber threats. The existing literature, for example [50], corroborates these findings, emphasizing the importance of robust and adaptable anomaly detection mechanisms in maintaining the security and integrity of critical infrastructure.

5.6.4. Isolation Forest

Isolation forest (IF) is an anomaly detection-focused unsupervised machine learning technique. Unlike traditional classification or regression algorithms, choosing a feature at random and then choosing a split value between the maximum and minimum values of that feature is how isolation forests work. This isolates observations. The key insight is that because anomalies are rare and distinctive, isolation is more likely to occur. This process isolates anomalies with fewer splits compared to normal points, which require more splits to be isolated. Moreover, isolation forests are different from Random Forests in their approach to constructing trees. While Random Forests are an ensemble of decision trees used for classification and regression tasks by aggregating the results of individual trees, isolation forests focus solely on isolating anomalies. The trees in an isolation forest are specifically constructed to isolate points, making them shallow and efficient.

For this study, the isolation forest was configured with 50 trees, a feature subset size of 0.6, a sample size of 0.6, and a contamination factor of 0.1. These hyperparameters (Table 11 with selected parameters in bold) were chosen to balance detection performance and computational efficiency.

Table 11. Hyperparameter set for isolation forest.

Hyperparameters	Values
Trees	**50**, 100, 200
Features	1.0, 0.8, **0.6**
Samples	auto, **0.6**, 0.8
Contamination	**0.1**, 0.2, 0.3

The performance metrics for the isolation forest on the SWaT dataset show an overall accuracy of 83%, indicating a reasonable ability to differentiate between normal and anomalous events. The precision for detecting normal events was high at 0.89, reflecting the model's capability to accurately identify true positives. However, the precision for anomalies was significantly lower at 0.36, suggesting a considerable number of false positives. Similarly, the recall for normal events was strong at 0.92, but the recall for anomalies was only 0.29, indicating that the model missed a substantial proportion of actual anomalies. The F1 score for normal events was robust at 0.90, but for anomalies, it was relatively low at 0.32. Additionally, the ROC curve area of 0.60 indicates that the model's ability to distinguish between normal and anomalous events is only moderately better than random guessing. Table 12 lists all the performance metrics for the IF. The technical performance of the isolation forest can be dissected further to understand these results. The algorithm's reliance on random partitioning to isolate anomalies inherently means that its effectiveness can vary significantly based on the dataset's characteristics. The SWaT dataset includes time-series data from a water treatment testbed, with both normal operations and several simulated attack scenarios. Normal data instances vastly outnumber the anomalies, which is typical for critical infrastructure datasets but poses a challenge for unsupervised learning algorithms.

Table 12. Performance of isolation forest.

True Positives (TP): 177	Accuracy: 0.8347
False Positives (FP): 319	Precision: 0.3614
True Negatives (TN): 3568	Recall: 0.2947
False Negatives (FN): 435	F1 Score: 0.3255

The lower precision and recall for anomalies suggests that while the isolation forest can identify many normal events accurately, it struggles with the detection of all instances of malicious activity. The severe class imbalance and absence of any prior knowledge or understanding of what malicious behavior may look like in terms of sensor readings also worsen the performance of the algorithm. The model's decision boundaries, defined by the randomly generated splits, may not capture these subtle deviations effectively, leading to missed detections (false negatives) and incorrect classifications of normal data as anomalies (false positives). Figure 13 shows the Confusion matrix for IF and Figure 14 shows its ROC characteristics.

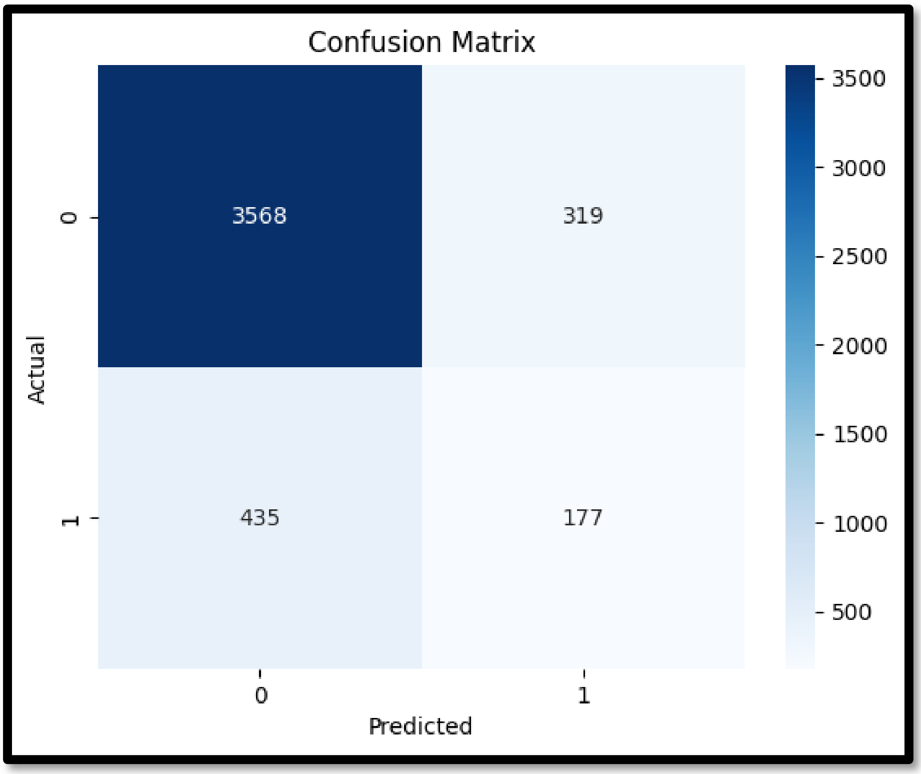

Figure 13. Confusion matrix of isolation forest.

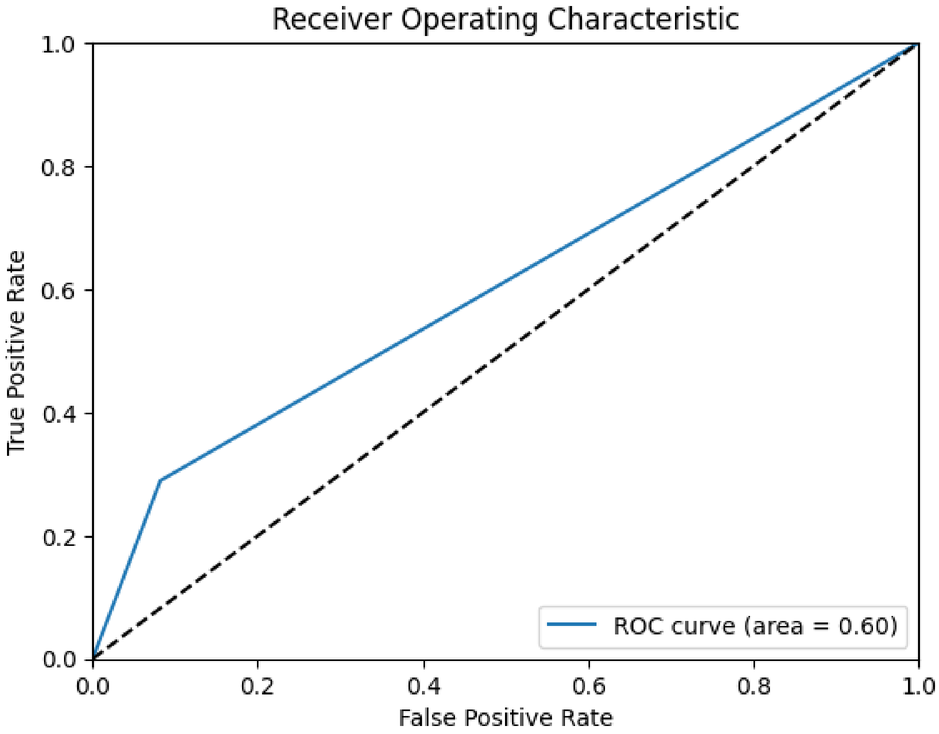

Figure 14. ROC characteristics for isolation forest.

Improving the performance of the isolation forest on such datasets might involve increasing the number of trees to provide a more nuanced partitioning of the feature space, as per the research by [51], although this comes at the cost of increased computational resources. Adjusting the contamination parameter, which estimates the proportion of anomalies in the data, can also impact performance. In scenarios like SWaT, where anomalies are rare, setting a lower contamination factor might help the model focus more on rare events.

Thus, while the isolation forest provides a reasonably decent method for unsupervised anomaly detection in critical infrastructure, albeit with plenty of room for improvement, its performance in detecting rare and subtle anomalies highlights the need for continuous refinement and combination with other detection techniques. This approach ensures comprehensive coverage and enhances the resilience of critical infrastructure systems against evolving threats. The existing literature, such as the research by [52], supports these findings, emphasizing the balance between computational efficiency and the need for nuanced detection mechanisms in critical infrastructure. Table 13 shows a summary of the reviewed algorithms with the key performance metrics examined: Random Forest emerges as a good candidate for anomaly detection in critical infrastructure.

Table 13. Summary of reviewed algorithms.

Algorithm	Accuracy	Precision	Recall	F1 Score
1D CNN	0.9671	0.8735	0.8981	0.8813
LSTM	0.9722	0.9639	0.8180	0.8850
Random Forest	0.9989	0.9930	0.9990	0.9960
One-Class SVM	0.8509	0.4785	0.5944	0.5261
Isolation Forest	0.8347	0.3614	0.2947	0.3255

6. Conclusions

This research comprehensively evaluated the effectiveness of machine learning using both supervised and unsupervised machine learning algorithms to detect anomalies within critical infrastructure, specifically using the SWaT dataset from the iTrust water treatment testbed. Among the supervised algorithms, Random Forest emerged as the most effective, showcasing robustness and high accuracy, essential for handling the complexity and variety of potential anomalies in such systems. One-dimensional convolutional neural network (1D CNN) and Long Short-Term Memory (LSTM) models also demonstrated strong performance, excelling in capturing spatial dependencies and effectively identifying temporal patterns. These attributes make supervised algorithms particularly suitable for anomaly detection in critical infrastructure, where the ability to process and analyze high-dimensional and temporal data is crucial.

Unsupervised algorithms, including isolation forest and one-class support vector machine (one-class SVM), provided valuable insights into anomaly detection without making use of labeled data. Isolation forest showed reasonable accuracy by isolating rare events through random partitioning, but its higher rates of false positives and negatives indicated a need for further refinement. One-class SVM, with its balanced performance, highlighted the importance of appropriate kernel selection to capture complex patterns in critical infrastructure data. One major challenge with unsupervised methods is their reliance on the assumption that anomalies are rare and distinct from normal data. In our dataset, the nature of attacks may not always produce clear, isolated anomalies. Subtle or sophisticated attacks might blend in with normal behavior, making it difficult for unsupervised algorithms to effectively distinguish between benign and malicious activities.

This can result in higher false-positive rates, where normal instances are incorrectly flagged as anomalies, or false negatives, where actual anomalies go undetected.

Overall, while supervised algorithms demonstrated superior accuracy and reliability due to their ability to leverage labeled data and capture intricate patterns, unsupervised algorithms offer a complementary approach by providing preliminary anomaly detection in scenarios where labeled data is scarce. The combination of both methods can enhance the robustness and resilience of intrusion detection systems in critical infrastructure, ensuring comprehensive coverage and the early detection of potential threats.

Combining the strengths of supervised and unsupervised algorithms can provide a more comprehensive detection mechanism. For instance, using unsupervised algorithms like isolation forest and one-class SVM to pre-filter data can help in identifying potential anomalies, which can then be analyzed more thoroughly using supervised models. This hybrid approach can reduce the chances of false positives and negatives, providing a more reliable detection system.

This study has several limitations that should be acknowledged. The results are based on the SWaT dataset, which, while representative, may not capture the full diversity of potential anomalies in all critical infrastructure systems. The dataset's specific characteristics, such as its focus on a water treatment plant, may limit the generalizability of the findings to other types of critical infrastructure. The dataset also has only 4 h of operational data, which could be a drawback in the performance of unsupervised learning methods such as one-class SVM and isolation forest, since the models may not have had enough data to learn the system behavior intricately and effectively.

The findings from this research offer several practical recommendations for industry professionals. The demonstrated effectiveness of Random Forest suggests it to be a prime candidate for anomaly detection in critical infrastructure. Integrating LSTM networks into IDSs can significantly enhance detection capabilities by capturing temporal patterns in the data. This is particularly important for detecting anomalies that develop gradually, ensuring timely intervention before significant damage occurs.

Overall, this research underscores the critical role of machine learning in enhancing the security and resilience of critical infrastructure systems. The application of machine learning algorithms, particularly those capable of handling both spatial and temporal data, provides a robust method for detecting and mitigating potential threats. The long-term impact of this research lies in its potential to inform the development of more advanced and adaptive IDS frameworks, ultimately contributing to the safeguarding of essential services and the prevention of disruptions caused by cyberattacks.

As the field evolves, continuous research and innovation will be crucial in addressing emerging threats and ensuring the security of critical infrastructure. The integration of machine learning into IDS frameworks represents a significant step forward in cybersecurity, offering the potential to detect and respond to threats more effectively and efficiently. This research provides a foundation for future studies, highlighting the need for ongoing refinement and the development of more sophisticated detection mechanisms to protect critical infrastructure from evolving cyber threats.

Author Contributions: Conceptualization, A.K. and J.A.G.; methodology, A.K.; validation, A.K.; investigation, A.K.; data curation, A.K.; writing—original draft preparation, A.K.; writing—review and editing, J.A.G.; visualization, A.K.; supervision, J.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zetter, K. Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid. Wired. Available online: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/ (accessed on 10 June 2024).
2. Aidan, J.S.; Verma, H.K.; Awasthi, L.K. Comprehensive Survey on Petya Ransomware Attack. In Proceedings of the 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS), Jammu, India, 11–12 December 2017; pp. 122–125. [CrossRef]
3. Zahra, S.R.; Ahsan Chishti, M. RansomWare and Internet of Things: A New Security Nightmare. In Proceedings of the 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 10–11 January 2019; pp. 551–555. [CrossRef]
4. Abdulova, E.; Kalashnikov, A. Categorization and Criticality Assessment of Facilities of Critical Infrastructure. In Proceedings of the 2022 15th International Conference Management of large-scale system development (MLSD), Moscow, Russia, 26–28 September 2022; pp. 1–5. [CrossRef]
5. Alimi, O.A.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S.; Alimi, K.O.A. A Review of Research Works on Supervised Learning Algorithms for SCADA Intrusion Detection and Classification. *Sustainability* **2021**, *13*, 9597. [CrossRef]
6. Malek, Z.S.; Trivedi, B.; Shah, A. User behavior Pattern -Signature based Intrusion Detection. In Proceedings of the 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 27–28 July 2020; pp. 549–552. [CrossRef]
7. Borkar, A.; Donode, A.; Kumari, A. A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS). In Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23–24 November 2017; pp. 949–953. [CrossRef]
8. What Is an Intrusion Detection System (IDS)? | IBM. Available online: https://www.ibm.com/topics/intrusion-detection-system (accessed on 19 April 2023).
9. Cazorla, L.; Alcaraz, C.; Lopez, J. Towards Automatic Critical Infrastructure Protection through Machine Learning. In *Critical Information Infrastructures Security*; Luiijf, E., Hartel, P., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2013; pp. 197–203. [CrossRef]
10. Sarker, I.H. *AI-Driven Cybersecurity and Threat Intelligence*; Springer: Cham, Switzerland, 2024; Available online: https://link.springer.com/book/10.1007/978-3-031-54497-2 (accessed on 12 July 2024).
11. Neshenko, N.; Bou-Harb, E.; Furht, B. A behavioral-based forensic investigation approach for analyzing attacks on water plants using GANs. *Forensic Sci. Int. Digit. Investig.* **2021**, *37*, 301198. [CrossRef]
12. Sharma, R.; Sharma, N.; Sharma, A. Application of Machine Leaning for Intrusion Detection in Internet of Things. In *Big Data Analytics in Intelligent IoT and Cyber-Physical Systems*; Springer: Singapore, 2024; Available online: https://link.springer.com/chapter/10.1007/978-981-99-4518-4_6?cv=1&code=67c3e40a-ea8c-4853-9451-3adc28ef1d80 (accessed on 14 July 2024).
13. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
14. Wongkaew, W.; Muanyoksakul, W.; Ngamkhanong, C.; Sresakoolchai, J.; Kaewunruen, S. Data driven machine learning prognostics of buckling failure modes in ballasted railway track. *Discov. Appl. Sci.* **2024**, *6*, 212. [CrossRef]
15. Yadav, G.; Paul, K. Architecture and security of SCADA systems: A review. *Int. J. Crit. Infrastruct. Prot.* **2021**, *34*, 100433. [CrossRef]
16. Santos, V.F.; Albuquerque, C.; Passos, D.; Quincozes, S.E.; Mossé, D. Assessing Machine Learning Techniques for Intrusion Detection in Cyber-Physical Systems. *Energies* **2023**, *16*, 6058. [CrossRef]
17. Pinto, A.; Herrera, L.-C.; Donoso, Y.; Gutierrez, J.A. Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors* **2023**, *23*, 2415. [CrossRef]
18. Elnour, M.; Meskin, N.; Khan, K.; Jain, R. A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems. *IEEE Access* **2020**, *8*, 36639–36651. [CrossRef]
19. Wang, M.; Yang, N.; Guo, Y.; Weng, N. Learn-IDS: Bridging Gaps between Datasets and Learning-Based Network Intrusion Detection. *Electronics* **2024**, *13*, 1072. [CrossRef]
20. Ali, A.; Naeem, S.; Anam, S.; Ahmed, M. Machine Learning for Intrusion Detection in Cyber Security: Applications, Challenges, and Recommendations. *Innov. Comput. Rev.* **2023**, *2*, 42–64. [CrossRef]
21. Raman, G.; Ahmed, C.; Mathur, A. Machine learning for intrusion detection in industrial control systems: Challenges and lessons from experimental evaluation. *Cybersecurity* **2021**, *4*, 27. [CrossRef]
22. Otoum, S.; Kantarci, B.; Mouftah, H. A Comparative Study of AI-based Intrusion Detection Techniques in Critical Infrastructures. *arXiv* **2020**, arXiv:2008.00088. [CrossRef]
23. Nader, P.; Honeine, P.; Beauseroy, P. Intrusion Detection in SCADA Systems Using One-Class Classification. In Proceedings of the 21th European Conference on Signal Processing (EUSIPCO), Marrakech, Morocco, 9–13 September 2013; pp. 1–5. Available online: https://hal.science/hal-01966009 (accessed on 14 July 2024).

24. Begli, M.; Derakhshan, F.; Karimipour, H. A Layered Intrusion Detection System for Critical Infrastructure Using Machine Learning. In Proceedings of the 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–14 August 2019; pp. 120–124. [CrossRef]
25. Kumar, A.; Sharma, I. Real-Time Threat Detection in Critical Infrastructure with Machine Learning and Industrial Control System Data. In Proceedings of the 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 29–31 December 2023; pp. 1–6. [CrossRef]
26. Al-Dhaheri, M.; Zhang, P.; Mikhaylenko, D. Detection of Cyber Attacks on a Water Treatment Process. *IFAC-PapersOnLine* **2022**, *55*, 667–672. [CrossRef]
27. Kravchik, M.; Shabtai, A. Detecting Cyberattacks in Industrial Control Systems Using Convolutional Neural Networks. *arXiv* **2018**, arXiv:1806.08110. [CrossRef]
28. Perales Gómez, Á.L.; Fernández Maimó, L.; Huertas Celdrán, A.; García Clemente, F.J. MADICS: A Methodology for Anomaly Detection in Industrial Control Systems. *Symmetry* **2020**, *12*, 1583. [CrossRef]
29. Haylett, G.; Jadidi, Z.; Nguyen, K. System-Wide Anomaly Detection of Industrial Control Systems via Deep Learning and Correlation Analysis. In *Artificial Intelligence Applications and Innovations*; Springer: Cham, Switzerland, 2021; pp. 362–373. [CrossRef]
30. 1999 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory. Available online: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset (accessed on 11 June 2024).
31. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 1 June 2024).
32. iTrust Labs_SWaT. iTrust. Available online: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_swat/ (accessed on 12 June 2024).
33. Wang, Z.; Li, W.; Tang, Z. Enhancing the genomic prediction accuracy of swine agricultural economic traits using an expanded one-hot encoding in CNN models. *J. Integr. Agric.* **2024**. [CrossRef]
34. Altameemi, Y.; Altamimi, M. Thematic Analysis: A Corpus-Based Method for Understanding Themes/Topics of a Corpus through a Classification Process Using Long Short-Term Memory (LSTM). *Appl. Sci.* **2023**, *13*, 3308. [CrossRef]
35. ArunKumar, K.E.; Kalaga, D.V.; Kumar, C.M.S.; Kawaji, M.; Brenza, T.M. Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends. *Alex. Eng. J.* **2022**, *61*, 7585–7603. [CrossRef]
36. Lu, X.-Q.; Tian, J.; Liao, Q.; Xu, Z.-W.; Gan, L. CNN-LSTM based incremental attention mechanism enabled phase-space reconstruction for chaotic time series prediction. *J. Electron. Sci. Technol.* **2024**, *22*, 100256. [CrossRef]
37. Akinpelu, S.; Viriri, S.; Adegun, A. Lightweight Deep Learning Framework for Speech Emotion Recognition. *IEEE Access* **2023**, *11*, 77086–77098. [CrossRef]
38. He, Z.; Wang, J.; Jiang, M.; Hu, L.; Zou, Q. Random subsequence forests. *Inf. Sci.* **2024**, *667*, 120478. [CrossRef]
39. Saheed, Y.K.; Abdulganiyu, O.H.; Majikumna, K.U.; Mustapha, M.; Workneh, A.D. ResNet50-1D-CNN: A new lightweight resNet50-One-dimensional convolution neural network transfer learning-based approach for improved intrusion detection in cyber-physical systems. *Int. J. Crit. Infrastruct. Prot.* **2024**, *45*, 100674. [CrossRef]
40. Inoue, J.; Yamagata, Y.; Chen, Y.; Poskitt, C.M.; Sun, J. Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 1058–1065. [CrossRef]
41. Abedzadeh, N. Implementing a New Algorithm to Balance and Classify the Imbalanced Intrusion Detection System Datasets. Ph.D. Thesis, The Catholic University of America, Wasignton, DC, USA, 2024. Available online: https://www.proquest.com/docview/2901755853/abstract/6BC430BBB7D54C83PQ/1 (accessed on 16 July 2024).
42. Piccininni, M.; Wechsung, M.; Van Calster, B.; Rohmann, J.L.; Konigorski, S.; van Smeden, M. Understanding random resampling techniques for class imbalance correction and their consequences on calibration and discrimination of clinical risk prediction models. *J. Biomed. Inform.* **2024**, *155*, 104666. [CrossRef]
43. Mazzanti, S. Your Dataset Is Imbalanced? Do Nothing! Available online: https://medium.com/data-science/your-dataset-is-imbalanced-do-nothing-abf6a0049813 (accessed on 16 June 2025).
44. Estévez, V.; Mattbäck, S.; Boman, A.; Liwata-Kenttälä, P.; Björk, K.-M.; Österholm, P. Acid sulfate soil mapping in western Finland: How to work with imbalanced datasets and machine learning. *Geoderma* **2024**, *447*, 116916. [CrossRef]
45. Zhao, X.; Zhang, L.; Cao, Y.; Jin, K.; Hou, Y. Anomaly Detection Approach in Industrial Control Systems Based on Measurement Data. *Information* **2022**, *13*, 450. [CrossRef]
46. Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. *arXiv* **2016**, arXiv:1607.00148. [CrossRef]
47. Kim, K.; Jeong, J. Real-Time Monitoring for Hydraulic States Based on Convolutional Bidirectional LSTM with Attention Mechanism. *Sensors* **2020**, *20*, 7099. [CrossRef]

48. Esmaily, J.; Moradinezhad, R.; Ghasemi, J. Intrusion detection system based on Multi-Layer Perceptron Neural Networks and Decision Tree. In Proceedings of the 2015 7th Conference on Information and Knowledge Technology (IKT), Urmia, Iran, 26–28 May 2015; pp. 1–5. [CrossRef]
49. Boukraa, L.; Essahraui, S.; Makkaoui, K.E.; Ouahbi, I.; Esbai, R. Intelligent Intrusion Detection in Software-Defined Networking: A Comparative Study of SVM and ANN Models. *Procedia Comput. Sci.* **2023**, *224*, 26–33. [CrossRef]
50. Aboah Boateng, E.; Bruce, J.W.; Talbert, D.A. Anomaly Detection for a Water Treatment System Based on One-Class Neural Network. *IEEE Access* **2022**, *10*, 115179–115191. [CrossRef]
51. Liu, T.; Zhou, Z.; Yang, L. Layered isolation forest: A multi-level subspace algorithm for improving isolation forest. *Neurocomputing* **2024**, *581*, 127525. [CrossRef]
52. Al Farizi, W.S.; Hidayah, I.; Rizal, M.N. Isolation Forest Based Anomaly Detection: A Systematic Literature Review. In Proceedings of the 2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE), Semarang, Indonesia, 23–24 September 2021; pp. 118–122. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.