Tabriz Journal of Electrical Engineering (TJEE), vol. 56, no. 1, 2026 Serial no. 115
DOI: 10.22034/tjee.2026.67140.5017

Design of a Hybrid LSTM-DT Intrusion Detection System in SCADA Networks

Majid Naghibian [1], Alireza Faraji [1*]

[1], Faculty of Electrical and Computer Engineering, University of Kashan, Kashan, Iran.

mnaghibian80@gmail.com
*arfaraji@kashanu.ac.ir

Received: 04/05/2025, Revised:06/11/2025, Accepted:30/05/2026.

Abstract
SCADA systems are critical infrastructures for managing and monitoring industrial processes, essential for controlling industrial operations. With the growing prevalence of cyber threats, detecting attacks on these systems poses a significant challenge. This study presents a hybrid model utilizing machine learning and deep learning techniques to detect cyber-attacks in SCADA networks. The proposed model integrates Long Short-Term Memory (LSTM) neural networks and Decision Tree (DT) models, trained on real industrial network traffic data. The hybrid model effectively detects intrusions with high accuracy, precision, recall, and F1-score, surpassing other approaches such as KNN and LSTM-CNN. Its superior ability to analyse network data and identify temporal patterns ensures robust performance. Additionally, the model has been validated in operational and real-time scenarios, demonstrating practical applicability. This research enhances SCADA system security and provides a framework for leveraging advanced machine learning models in industrial cybersecurity.

Keywords
Industrial Control Systems, Cyberattacks, Long Short-Term Memory, Decision Tree, Intrusion Detection System.

1. Introduction

SCADA systems serve as critical infrastructures in industries such as power plants, refineries, water treatment facilities, and transportation networks, playing a pivotal role in controlling and monitoring industrial processes. With the increasing integration of these systems into networked environments and their exposure to the internet, they have become prime targets for cyberattacks. These attacks, often executed through communication networks or external access points, can disrupt operations, cause financial losses, and even lead to catastrophic failures in critical infrastructures. As a result, ensuring the security of SCADA systems has become one of the top priorities in cybersecurity research.

Traditional Intrusion Detection Systems (IDS) have been widely used to identify and mitigate threats by analysing network traffic and detecting anomalies. However, these systems face significant challenges, including high false-positive rates, limitations in processing large datasets, and difficulties in adapting to evolving attack patterns. To address these shortcomings, machine learning and deep learning techniques have emerged as powerful tools for enhancing threat detection [1]. Recent studies, such as references [2-4], have explored various algorithms like K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Decision Trees for detecting cyberattacks in SCADA systems. These approaches leverage labeled data and behavioral analysis to identify both known and unknown attack patterns, demonstrating promising results.

In another study [5], a testbed was designed to evaluate machine-based detection systems. In this study, models such as KNN, Naïve Bayes (NB), Logistic Regression (LR), Random Forest (RF), and Decision Trees were compared. [6] It also reviews different algorithms for detecting cyberattacks in SCADA systems. The research results show that the decision tree in particular C5.0 has a better performance in intrusion detection compared to other models. It is able to identify the detected patterns using assigned labels and behavioral analysis. In addition [7, 8] , reference proposed a deep learning-based approach using advanced models like SVM, KNN, and ANN to detect cyberattacks at lower levels of the OSI model, specifically Layers 1 and 2 of Modbus networks.

Recurrent neural networks (RNNs), particularly LSTM models, have also been utilized to address complex threats like data injection attacks in industrial control systems [9].These models predict temporal sequences and establish correlations between discrete data points, enabling the detection of stealthy attacks. However, challenges such as delays in detecting fast-paced attacks remain. Reference [10] combined LSTM and RNN models, achieving improved detection of complex attacks such as DOS, R2L, U2R, and PortScan using the KDD Cup 1999 dataset.

Despite advancements, current systems often struggle with dynamic network environments and sophisticated attack vectors. For instance, reference [11] emphasized the need for customized rules tailored to SCADA systems, particularly for detecting start/stop attacks on Programmable Logic Controllers (PLCs). Similarly,

reference [12] explored the use of SVM to identify abnormal traffic but noted challenges in handling complex attacks and networks with dynamic IP addresses. Two recent studies further contribute to the field by proposing innovative hybrid approaches. In the first study [13], Najjar and Moattar introduced a hybrid intrusion detection system combining Hidden Markov Models (HMM) and Extreme Learning Machines (ELM). Their approach involves preprocessing network traffic data, training an HMM using the Baum-Welch algorithm, and applying the Viterbi algorithm to extract the most probable state sequences. These sequences are then fed into an ELM classifier to categorize data as normal or attack traffic. Using the DARPA98 dataset, the authors demonstrated that their method achieves higher accuracy and lower false-positive rates compared to previous approaches. This hybrid model highlights the potential of integrating statistical and machine learning techniques for enhanced intrusion detection. In another study [14],Ghasabi and Deypir proposed a novel method for detecting and mitigating Distributed Denial-of-Service (DDoS) attacks in Software-Defined Networks (SDNs). They utilized the Jeffery distance metric, a statistical technique, to identify abnormal traffic patterns indicative of DDoS attacks. The authors leveraged the unique architecture of SDNs to implement their algorithm, which was evaluated using the Mininet simulator in a Linux environment. Their results showed that the proposed method outperforms existing techniques in terms of efficiency and effectiveness, providing a robust solution for securing SDN-based infrastructures against distributed attacks.

Building on prior research, this paper introduces a hybrid intrusion detection system that combines LSTM neural networks with Decision Trees DT. The proposed model analyses network packets, extracts key features, and leverages the strengths of both sequential data processing and rapid decision-making. This approach not only improves detection accuracy but also reduces false alarms, offering a robust solution for securing SCADA systems against evolving cyber threats. In this research, an attempt has been made to propose a model that integrates machine learning and neural network methods. The model is trained on data extracted from network information and is designed to identify known attacks. The primary goal of this study is to design a system that can effectively function as an intrusion detection system in a real industrial network.

2. Research Methodology

Intrusion detection systems play a key role as one of the most effective security tools for identifying and countering cyberattacks in industrial control systems and SCADA infrastructures. These systems, by comprehensively analysing network traffic and system activities, identify unauthorized attempts to access, manipulate, or disable critical infrastructures and provide reports for corrective actions. Intrusion detection methods are generally divided into two main categories: signature-based and anomaly-based. Signature-based methods rely on predefined rules to detect known attacks, while anomaly-based methods identify abnormal activities by comparing system behaviour with standard models.

Intrusion detection systems can be classified into three categories based on their location in the network:

Network-Based Intrusion Detection Systems (NIDS): These systems monitor all network traffic.

Host-Based Intrusion Detection Systems (HIDS): These systems analyse the activities of specific hosts.

Distributed Intrusion Detection Systems (DIDS): These systems combine the features of the previous two types and send reports to a central station.

Advanced methods, including machine learning and deep learning techniques, have been developed to enhance intrusion detection capabilities. These systems use mathematical models to identify both known and unknown attacks. By analysing extensive datasets and creating learning models, they contribute to improving accuracy and reducing false alarms [15, 16].

Machine learning and deep learning models require proper training to achieve optimal and acceptable performance. This process necessitates access to a comprehensive and sufficient knowledge base. In this research, the required knowledge base was extracted from captured data in a real industrial control network. Unlike many studies that utilize pre-existing datasets for model training, this approach ensures better alignment of the model with the target network's characteristics and improves its recognition of real-world conditions.

In the following sections, data collection methods and the preprocessing steps for supervised learning using labelled data will be discussed in detail.

3. Data Collection and Preparation

In this research, the data required for training and evaluating the proposed models were obtained from a real industrial control network. These data include network traffic collected in an experimental environment using relevant hardware and software. The experimental setup includes industrial control equipment such as Siemens S7-1200 PLCs, HMI devices, and WinCC monitoring software, as well as attack simulation tools like the Kali Linux system. The structure of the experimental environment is depicted in Figure 1. The collected data consist of network packets transmitted via external ports over TCP/IP protocols. Since the attacks investigated in this study are conducted on industrial network controllers, only data related to the transport layer and the TCP protocol have been considered.

The software Wireshark, one of the advanced tools for monitoring and analyzing network traffic, was used to capture and analyze data. This software is capable of decoding network packets based on their protocols and can extract useful information such as IP and MAC addresses, port numbers, packet lengths, and data contained in each packet. However, due to the point-to-point structure of the network, Wireshark cannot access all incoming packets. To address this limitation, the Port Mirroring feature was utilized. This capability allows all incoming packets to a specific port of the control system to be forwarded, enabling Wireshark to comprehensively monitor and record network traffic [17].

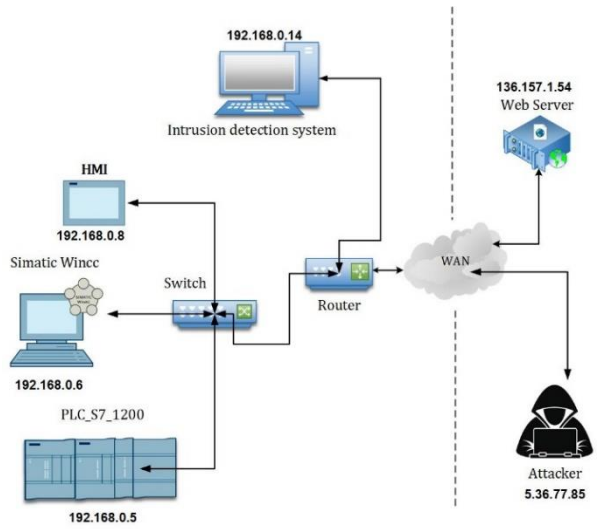

Fig.1. General layout of the laboratory space.

After capturing the data, network packets were filtered based on specific criteria to remove irrelevant information related to the control system. This process involves isolating packets that have attributes relevant to the controller, such as the MAC address, source IP, controller port numbers, open ports, and key request data. Irrelevant or redundant data, such as frequently changing MAC addresses, were excluded. After this step, the extracted information was used to create the knowledge base required for supervised training of the models. Specifically, after executing attacks in the experimental environment, attack-related packet data were compared with normal data to identify key changes and extract important features for detecting attacks. The prepared data was then used to train and evaluate machine learning and deep learning models.

4. Data Preprocessing

Data preparation and preprocessing are critical steps before training machine learning and deep learning models, as they directly impact the accuracy and performance of these models [18]. In this research, the data collected from the real industrial control network were first examined for quality, and preprocessing steps were applied. These steps included normalization and standardization of the data to harmonize the scale of features and facilitate the model training process. Normalization scales feature values to the range [0,1], while standardization adjusts the data to have a mean of zero and a standard deviation of one. These methods improve the performance and accelerate the learning process, especially for scale-sensitive models like neural networks. Additionally, outliers were identified and removed prior to applying these processes to prevent their negative impact on the results.

Furthermore, textual categorical data were converted into numerical values so that machine learning models could process them. For this purpose, techniques such as label encoding and one-hot encoding were used, ensuring that categorical values did not introduce any incorrect ordering. The data were then divided into three sets: training, validation, and testing, allowing the model to evaluate its performance on unseen data. Typically, 60–80% of the data were allocated for training, 10–20% for validation, and 10–20% for testing. All these processes were implemented using Python and tools available in libraries such as Scikit-learn, including modules like Label Encoder and Min Max Scaler. These steps transformed the data into a standardized format and provided the necessary conditions for accurate model training.

5. Design of Cyberattacks

In this research, to create a knowledge base for training and evaluating the performance of the models, some common attacks in the SCADA systems domain were designed and implemented. The selected attacks include Distributed Denial of Service (DDoS), Start/Stop attacks, and Port Scanning attacks, which are introduced into the system through the network transport layer. Each attack was implemented with a specific goal, and the characteristics of their associated data were analysed and evaluated to improve the accuracy of the intrusion detection system.

Distributed Denial of Service Attack:

This attack floods the target system with a high volume of requests, saturating server resources and causing service disruptions. Two common types of this attack, SYN Flood and HTTP Flood, were examined. The SYN Flood attack occupies system resources by sending half-open packets at the transport layer, while the HTTP Flood attack targets web servers by sending multiple requests at the application layer. The impact of these attacks on control systems can lead to process failures and significant damages.

Start/Stop Attack:

This type of attack directly disrupts the operation of the control system and can cause physical damage to equipment through repeated execution. The structure of the network packets and the identification of control-related data for this attack were used to design appropriate patterns and evaluate the intrusion detection system. The Metasploit framework was employed to simulate these attacks.

Port Scanning Attack:

This attack aims to identify open ports and vulnerabilities in the system. Methods such as SYN Scan and FIN Scan were used for this attack. Characteristics of the identified packets, such as the maximum segment size (MSS), the high number of requests sent to network ports, and the types of responses received, were used as criteria for detecting these attacks. The MSS value represents the maximum amount of data that can be transmitted in a single network connection and is calculated using the following formula:

$$MSS = MTU - (TCP\ header + IP\ header)$$

After designing and executing each attack, the corresponding packets were captured, and changes in the impactful features of the packet state were analysed. Features with the highest correlation coefficients were selected as the necessary knowledge for training the

model. The correlation coefficients of the analysed features are shown in Figure 2.

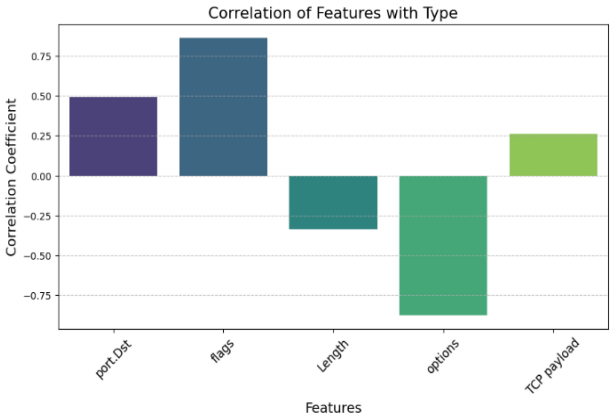

Fig. 2. Correlation value of features with package type.

6. Model Design and Training

To detect cyberattacks in industrial control systems, the selected model should have criteria such as detection accuracy, learning speed, and the ability to analyse sequential data. These features are critical for SCADA systems that are very sensitive to latency. In this research, a hybrid model combining short-term memory neural networks and decision trees was used. The LSTM network was selected due to its ability to analyse time-dependent data and dynamic behaviours, while the decision tree was selected due to its fast-decision-making ability and high interpretability [19].

The hybrid model is evaluated in three configurations:

1. First configuration: LSTM network is used to extract temporal features and its output is passed to the decision tree for final decision making.
2. Second configuration: The decision tree extracts important features, while the LSTM network takes on the role of the final decision maker.
3. Third configuration: Both models operate in parallel in feature extraction, and their outputs are combined in the last layer and decisions are made about the data state with the help of activation functions.

These three configurations are designed to improve accuracy, speed, and interpretability.

To train the model, after data preparation, key settings are adjusted, including the selection of the number of layers, activation functions, partitioning, and optimization algorithms. In the LSTM model, the ReLU activation function is used for the initial and hidden layers and the SoftMax function is used for the output layer. This selection is made to increase the learning speed in the hidden layers and address the multi-class nature of the problem. In addition, the Adam optimizer and the cross-class entropy loss function are used to reduce the model error.

In the decision tree, the Gini criterion is used for splitting, and depth, minimum leaf size, and branch constraints are used to prevent overfitting. The training process is performed in three defined configurations, and the models are trained with real and labelled data in a supervised manner. Advanced techniques such as Dropout are used to prevent overfitting, and appropriate parameter settings, such as the number of training sessions and batch size, are adjusted to optimize the model's performance. These settings ensure that the model can accurately detect both known and unknown attacks in a real-world industrial environment.

7. Evaluation of the Proposed System

After training the model, its performance is evaluated on the test data to ensure that the model has been trained correctly and possesses good generalization capabilities. This evaluation is conducted based on metrics such as accuracy, precision, recall, F1-score, and other relevant criteria to validate the model. The evaluation results for the model in all three configurations are presented in Table I.

Table I. Evaluation Results of the Model Validation in Three Configurations

Dataset	Accuracy	Precision	Recall
First Configuration	98.43%	98.80%	99.07%
Second Configuration	90.99%	94.00%	95.10%
Third Configuration	99.45%	99.23%	99.49%

Another key tool for analysing the performance of the model is the confusion matrix, which consists of four distinct components:

- True Positive (TP): Cases where the model correctly identifies positives.
- True Negative (TN): Cases where the model correctly identifies negatives.
- False Positive (FP): Cases where the model incorrectly identifies positives.
- False Negative (FN): Cases where the model incorrectly identifies negatives.

In Figure 3, a comparison of all three configurations of the proposed model is presented using the confusion matrix metric.

In the evaluation of the models, the hybrid model combining LSTM and Decision Tree is analysed in three different configurations.

- First Configuration: The LSTM is used for sequential data analysis, and the Decision Tree is employed for decision-making. This leads to high performance of the LSTM in identifying complex patterns.
- Second Configuration: The Decision Tree extracts feature and then sends them to the LSTM. However, this results in reduced accuracy due to the diminished depth of sequential data.
- Third Configuration: Both models operate in parallel, yielding optimal results in terms of accuracy, the ability to analyse complex patterns, and interpretable features.

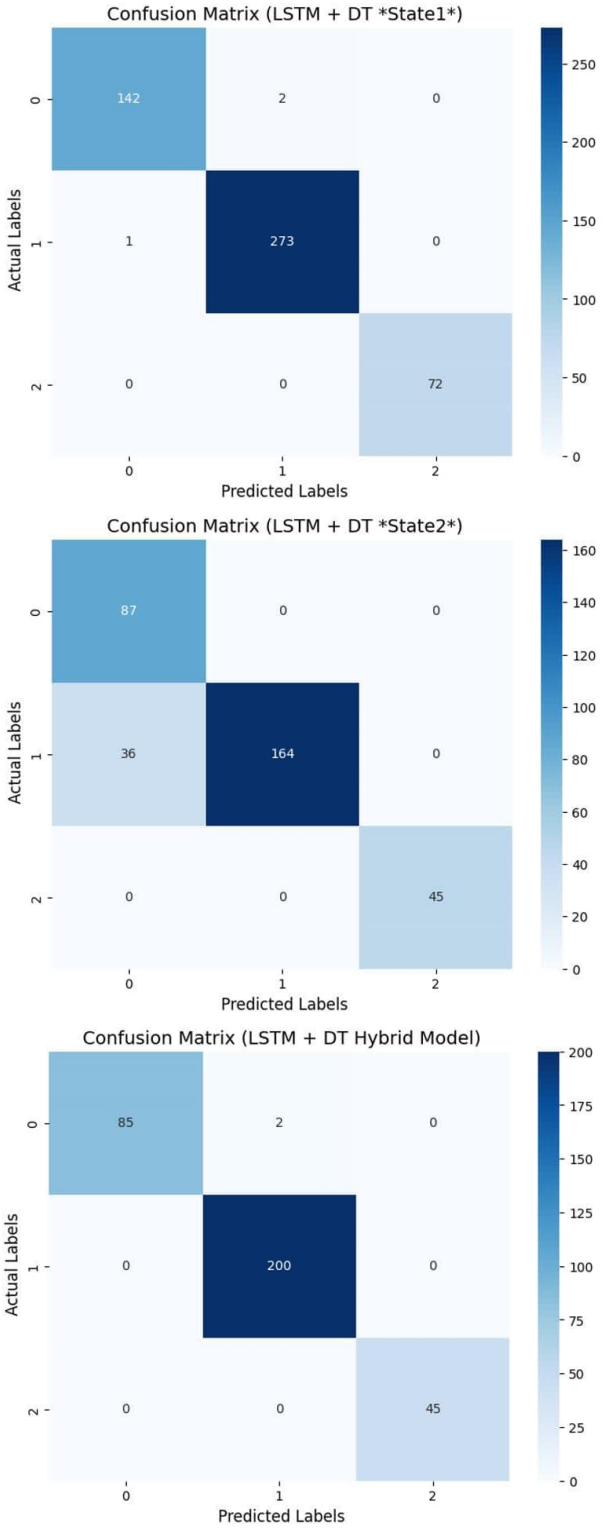

Fig. 3. Confusion matrices.

Table II: Evaluation Results of the Models

Model	Accuracy	Precision	F1-Score
State 1	98.43%	98.16%	98.11%
State 2	90.99%	89.16%	90.24%
State 3	99.45%	99.40%	99.67%
LSTM	95.00%	98.11%	97.62%
DT	100%	100%	100%
LSTM-CNN	92.15%	91.28%	92.76%
SVM	86.20%	85.74%	90.21%
KNN	66.17%	62.31%	71.35%

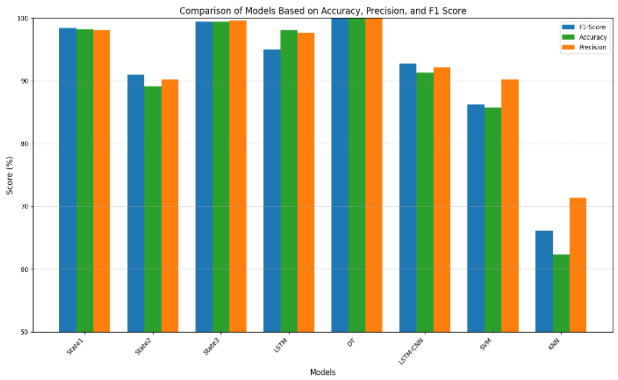

Fig. 4. Model comparison chart

Based on the comparative results of the hybrid model with other models, as shown in Figure 4 and Table 2, the LSTM-DT model demonstrates superior performance. Although the Decision Tree model achieves 100% detection accuracy in theory, its overfitting reduces its ability to generalize to new data in contrast, the LSTM-DT hybrid model in the first and third cases provides the best overall performance, with higher accuracy and less risk of overfitting.

In the practical evaluation, the model was implemented in a real network and demonstrated satisfactory performance in identifying attacks with minor pattern variations. This assessment showed that the hybrid model is capable of handling diverse scenarios, whereas other models lose accuracy when faced with minor changes. (Figure 5)

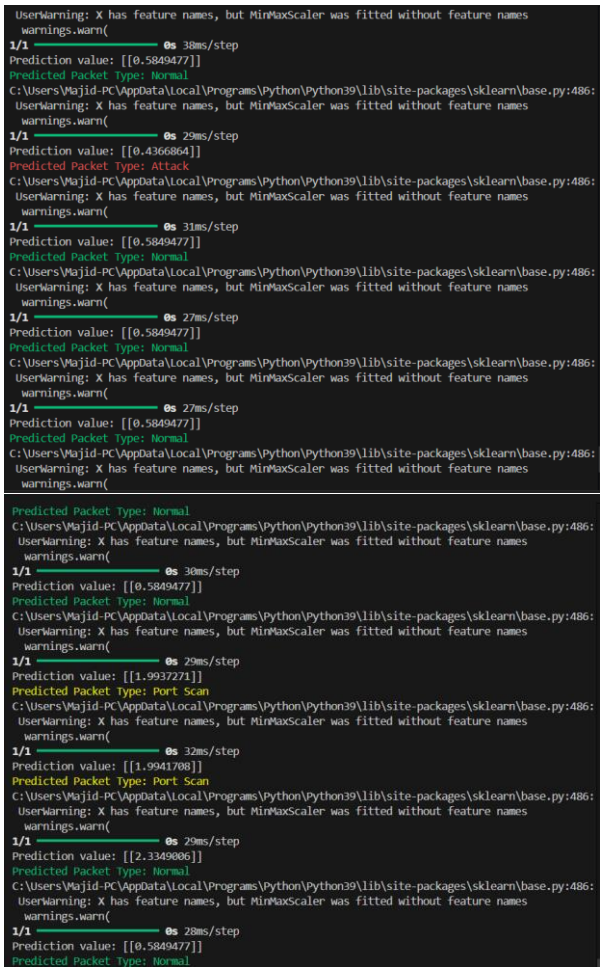

Fig. 5. IDS system diagnosis results

8. Conclusion

This research designed and implemented a hybrid system based on machine learning and deep learning for detecting cyberattacks in SCADA networks. The LSTM-DT model, leveraging its ability to process complex data, demonstrated a low false detection rate and resistance to overfitting, offering optimal performance compared to other models. Evaluations conducted in a real-world environment showed that the system successfully identified cyber threats quickly and accurately, exhibiting a high capability to ensure the cybersecurity of industrial infrastructures.

In future research, focusing on other network layers and widely used protocols could lead to the design of a more comprehensive system for attack detection. Analysing layers such as the application layer and protocols like DNP3 can help identify new vulnerabilities. Additionally, developing the model for online and self-supervised learning could enhance its effectiveness in addressing emerging attacks. Furthermore, the use of distributed systems in multi-domain networks and evaluating the model's capability in detecting more sophisticated attacks, such as FDI (False Data Injection) and MITM (Man-in-the-Middle), could outline future research directions.

References

[1] Amanoul, S.V., et al., *Intrusion Detection Systems Based on Machine Learning Algorithms*, in *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*. 2021. p. 282-287.

[2] Mohamad Kaouk, J.-M.F., Marie-Laure Potet, Roland Groz, *A Review of Intrusion Detection Systems for Industrial Control Systems.* 2019.

[3] Mubarak, S., et al., *Anomaly Detection in ICS Datasets with Machine Learning Algorithms.* Computer Systems Science and Engineering, 2021. **37**(1): p. 33-46.

[4] Rocio Lopez Perez, F.A., Ridha Soua,Thomas Engel, *Machine Learning for Reliable Network Attack Detection in SCADA Systems.* 2018.

[5] Marcio Andrey Teixeira, T.S., , Maede Zolanvari, Raj Jain, Nader Meskin, Mohammed Samaka, *SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach.* Future Internet, 2018.

[6] Manish Kumar, D.M.H., Dr. T. V. Suresh Kumar, *Intrusion Detection System Using Decision Tree Algorithm.* 2012.

[7] Teixeira, M., et al., *SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach.* Future Internet, 2018. **10**(8).

[8] Wang, W., et al., *A stacked deep learning approach to cyber-attacks detection in industrial systems: application to power system and gas pipeline systems.* Cluster Comput, 2022. **25**(1): p. 561-578.

[9] Wei Wang, Y.X., Lu Ren, Xiaodong Zhu, Rui Chang, Qing Yin, *Detection of Data Injection Attack in Industrial Control System Using Long Short Term Memory Recurrent Neural Network.* 2018.

[10] Jihyun Kim, J.K., Huong Le Thi Thu, and Howon Kim, *Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection.* 2016.

[11] Sagarika Ghosh1, S.S., *A Survey of Security in SCADA Networks: Current Issues and Future Challenges.* IEEE Aerospace and Electronic Systems Magazine, 2019: p. 23.

[12] Terai, A., et al., *Cyber-Attack Detection for Industrial Control System Monitoring with Support Vector Machine Based on Communication Profile*, in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 2017. p. 132-138.

[13] س. معطر, تشخیص نفوذ شبکه با استفاده از رویکرد and .نجار, م ترکیبی مدل مخفی مارکوف و یادگیری ماشین مفرط. مجله مهندسی برق دانشگاه تبریز, ۲۰۱۹. ۴۸(4): p. 1807-1817.

[14] در *DDOS* م. دی‌پیر, تشخیص و کاهش اثر حملات and .قصابی, م شبکه‌های نرم‌افزار محور با استفاده از تکنیک فاصله جفری. مجله مهندسی برق دانشگاه تبریز, ۲۰۱۸. ۴۸(3): p. 1287-1300.

[15] Ahmad, Z., et al., *Network intrusion detection system: A systematic study of machine learning and deep learning approaches.* Transactions on Emerging Telecommunications Technologies, 2020. **32**(1).

[16] Alzahrani, A. and T.H.H. Aldhyani, *Design of Efficient Based Artificial Intelligence Approaches for Sustainable of Cyber Security in Smart Industrial Control System.* Sustainability, 2023. **15**(10).

[17] Arifin, M.A.S., et al., *Oversampling and undersampling for intrusion detection system in the supervisory control and data acquisition IEC 60870 - 5 - 104.* IET Cyber-Physical Systems: Theory & Applications, 2024. **9**(3): p. 282-292.

[18] Ferrag, M.A., M. Babaghayou, and M.A. Yazici, *Cyber security for fog-based smart grid SCADA systems: Solutions and challenges.* Journal of Information Security and Applications, 2020. **52**.

[19] Xin, Y., et al., *Machine Learning and Deep Learning Methods for Cybersecurity.* IEEE Access, 2018. **6**: p. 35365-35381.