Evaluation of Machine Learning Algorithms for Intrusion Detection in SCADA Systems

Maria Eduarda Cher Benetis dos Santos
São Paulo State Univ. of Tech. (FATEC)
Ourinhos, Brazil
maria.santos@fatecourinhos.edu.br

Fadir Salmen
São Paulo State Univ. of Tech. (FATEC)
Ourinhos, Brazil
fadir.salmen@fatecourinhos.edu.br

Thiago José Lucas
São Paulo State Univ. of Tech. (FATEC)
Ourinhos, Brazil
thiago@fatecourinhos.edu.br

Tiago Martins Ferreira
São Paulo State University (UNESP)
Bauru, Brazil
tiago.ferreira@unesp.br

Fernanda Mara Cruz
São Paulo State Univ. of Tech. (FATEC)
Ourinhos, Brazil
fernanda.cruz@unesp.br

Kelton Augusto Pontara da Costa
São Paulo State University (UNESP)
Bauru, Brazil
kelton.costa@unesp.br

***Abstract*—SCADA systems are widely used in critical industries such as energy and water for process monitoring and control. Due to their interconnection with communication networks, these systems are susceptible to various cyberattacks. This work seeks to detect attacks that could compromise the integrity and availability of SCADA systems, evaluating the performance of machine learning algorithms. Several tests were performed on the WUSTL-IIOT-2018 dataset in order to produce a comprehensive analysis of the performance of different classifiers. The results obtained demonstrated that the selected algorithms performed satisfactorily in the experiments carried out, highlighting their potential to strengthen the security of these critical systems**

***Keywords*—SCADA; Intrusion Detecion; Machine Learning.**

I. Introduction

SCADA (Supervisory Control and Data Acquisition) systems are fundamental in controlling and monitoring processes in various industries, enabling efficient and remote management of critical operations. For [1] since their emergence in the 1960s, these systems have evolved significantly, incorporating technological advances that have made them more sophisticated and comprehensive.

Inspired by the need for remote supervision and control of industrial processes, SCADA systems were originally designed as hardware solutions for monitoring and control. According to [2], over the years, these systems have adapted to the demands of an ever-evolving digital age, integrating with digital communication networks and adopting cloud computing concepts to improve their efficiency and reach. However, this increasing complexity and connectivity have also brought significant cybersecurity challenges. As they have become vital parts of critical infrastructure, SCADA systems have become potential targets for various cyberattacks.

With the advent of sophisticated and targeted attacks such as "Stuxnet", "BlackEnergy", "Triton" and the "Colonial Pipeline attack", the vulnerability of networks that support critical sectors such as energy, water, transportation and healthcare has been exposed. These incidents demonstrate malicious actors' ability to penetrate highly protected systems and reveal the catastrophic consequences that such attacks can trigger, from physical damage to large-scale socio-economic disruption.

Intrusion Detection Systems (IDS) play a vital role in this context. These specialized tools monitor network traffic for suspicious activity that could indicate the presence of a cyber attack.

By integrating machine learning techniques into IDSs, they become significantly better at identifying suspicious behavior and distinguishing between legitimate and potentially malicious traffic. The algorithms can learn from huge sets of network traffic data, detecting complex patterns that indicate intrusion activity. This approach enables more accurate and efficient detection of cyber threats, thus strengthening the security of networks and systems.

Given this scenario, the problem that guided this work was the following question: Can machine learning techniques in the training of intrusion detection systems contribute to an attack classification scenario that minimizes false positive and false negative events?

The general objective of this work was to investigate and evaluate the effectiveness of different machine learning algorithms to detect possible intrusions in environments that use SCADA systems to protect them against cyber threats, aiming to improve the security and resilience of these critical infrastructures. The specific objectives were:

- Investigate the most common vulnerabilities of SCADA systems that make them susceptible to cyber-attacks;
- Analysis of the WUSTL-IIOT-2018 dataset;
- Division of data into training and testing, ensuring a balanced division between classes;
- Implementation of the Random Forest, Decision Tree, KNN, Neural Network, Gradient Boosting and Naive Bayes algorithms to perform intrusion detection;
- Evaluation of the models using metrics such as Precision Recall and F1-Score, focusing on minimizing false positives and negatives and comparing the performance between the algorithms.

The main contributions of this work are: to encourage the scientific community to continue developing robust techniques for protecting SCADA systems, and to provide a comparative analysis of the performance of different algorithms for detecting attacks in SCADA environments.

II. Theoretical Background

This Section provides a comprehensive analysis of the current state of research related to SCADA attacks and intrusion detection strategies using machine learning. Throughout this Section, a detailed review of related work was prepared, highlighting important points that address specific vulnerabilities and notable incidents of past attacks. In addition, a machine learning-based IDS comparing different types of algorithms was developed.

A. Correlated work

In [3] the authors proposed the Measurement Intrusion Detection System (MIDS) as a solution to the challenges encountered in detecting attacks in industrial control systems. The MIDS can identify anomalous activities by analyzing measurement data from the SCADA system, even when attackers try to mask them. The application of machine learning techniques and testing on a HIL platform demonstrated the effectiveness of the MIDS in detecting stealthy attacks, highlighting the superior performance of Random Forest as a classifier algorithm, achieving an accuracy of 99.76%.

In [1] the authors conducted a study on intrusion detection in SCADA systems using real data from a Mississippi State University (MSU) gas pipeline system. In their research, they evaluated Machine Learning algorithms, including Random Forest (RF), Bidirectional Long Short Term Memory (BLSTM), and Support Vector Machine (SVM). The results revealed that both RF and BLSTM achieved exceptional F1 scores of over 99% and 96%, respectively, indicating significant effectiveness in intrusion detection.

In their study, [4] proposes a modified decision tree-fused chi-squared feature selection technique (Chi+MDT) for intrusion detection in SCADA networks. The proposed technique consists of three phases: data preparation, feature selection, and deployment of a modified Decision Forest for anomaly detection and classification. The accuracy achieved was 99.98%, 99.09%, 99.55%, and 99.91% on the WUSTL-IIoT-2018, ICS-SCADA, NSL-KDD, and CICDS2018 datasets. Furthermore, the technique was efficient regarding runtime, with values of 0.155s, 0.122s, 0.91s, and 0.32s on the respective datasets.

As show in [5], the authors developed an integrated framework for intrusion detection in SCADA power grids. This framework combines RFE-XGBoost-based feature selection techniques and majority voting ensemble methods. The accuracy of the majority vote-based ensemble method is higher and more accurate than the other nine classifiers, with an accuracy of around 98.24%.

In [6] the authors proposed an Ensemble Learning technique based on Decision Forest to detect distributed denial of service (DDoS) attacks in SCADA systems. Traffic data is collected in a simulated experimental network topology for training and testing the learning models. The performance optimization of the models is achieved through feature selection and hyperparameter tuning techniques. The experimental results indicate that the Decision Tree, Ensemble Boosted Trees, Ensemble Bagged Trees and Ensemble RUSBoost Trees techniques achieved an accuracy of 91.33%, 92.9%, 92.7% and 92.6%, respectively.

A generic framework for designing a machine learning-based Intrusion Detection System (IDS) for SCADA power systems is proposed and evaluated in [7]. In the offline training module, datasets from 15 different files were combined, augmented, and balanced with synthetic data generation using CT-GAN. Three classical machine learning algorithms (Decision Tree, Random Forest, and Gradient Boosting) were evaluated and compared using cross-validation and holdout validation. The models trained with augmented data showed superior performance, resulting in a more effective generic intrusion detection model. With feature selection (RFE-RF) applied to the 50 most significant features, the accuracy achieved was 93.97% on an unseen dataset.

In his work [8], he addresses the difficulty of detecting attacks targeting the physical processes of plants, such as Man-In-The-Middle (MITM), Replay and DoS (SYN flood) attacks. Using the Support Vector Machine (SVM) algorithm based on the fuzzy classifier in a non-parallel hyperplane, the method can classify complex datasets, such as Modbus/TCP traffic data and process state simulations, with a high accuracy of 93.73% in distinguishing between normal and abnormal data.

B. Intrusion Detection Systems

According to [9], actions related to maintaining the integrity, accessibility, or reliability of an electronic data source or a communication network define the objectives of an IDS. The authors provide a historical reading of the emergence and evolution of IDS, attributing to James Anderson - when the article "Computer Security Threat Monitoring and Surveillance"' was published, by [10], the first report of an IDS. Two notable events in the evolution of IDS occurred in 1986 and later in 1993 when publications scientifically documented intrusion detection models based on statistical network traffic analysis (both created by Dorothy Denning and Peter Neumann in [11]).

An IDS as software developed to detect attacks that have the potential to cause damage to a communication network or systems, whether they originate from an insecure medium such as the Internet or a local network [12]. Such software executes security countermeasures when it detects an anomaly in network traffic or hosts' behavior that may characterize an attack.

The authors highlight that many approaches have recently been applied to increase the detection rate, generally involving Machine Learning techniques (situations where the detector learns through training to detect anomalies), Rule Based (case where the detector has characteristic signatures of attacks and only compares real traffic with the signatures of already known attacks) and/or classical statistical methods (use of calculations of mean, standard deviation, median, among others).

IDSs have two classification dimensions: Network Topology and Packet Approach. The first dimension, according to [13], defines whether the detector works by analyzing the network flow (topologically, in this case, the IDS is normally placed in the communication network in the form of a bridge or by receiving a copy of all trafficked packets through mirroring of a port of the switch) or by analyzing the behavior of the operating system, a situation in which the detector works by observing processing measurements, disk space, memory usage or auditing records made in the system logs.

The second classification dimension concerns the way the detector analyzes the data. [14] states that in this categorization, there are two classes: signature detection and anomaly detection. The authors explain that in signature detection, the IDS has a database of known attacks, and its job is basically to compare the packets that arrive on the Network with its database to verify similarities. In anomaly detection, however, the system defines a model of what would be the "normal"' behavior of the Network or system, causing packets that are not classified as "normal" to be labeled as malicious.

C. Machine Learning

According to [15], machine learning, an area of artificial intelligence and computer science, focuses on using data and algorithms to replicate the human learning process, aiming to improve accuracy gradually. It is noteworthy that, applied to cybersecurity, machine learning makes malware detection more practical, scalable, and efficient compared to traditional methods that require human intervention. Machine learning involves creating and managing new patterns through algorithms, enabling real-time detection of active threats, and helping security teams proactively prevent breaches. This technology significantly impacts cybersecurity by facilitating various threat identification and mitigation techniques.

D. SCADA

According to [16], SCADA systems is essential for collecting, monitoring, and controlling real-time data from devices, sensors, and equipment in a network within critical infrastructures. Previously isolated, many of these systems are now connected to the Internet and corporate networks, increasing the operability and efficiency of industries.

As highlighted by [17], SCADA systems are composed of four main levels where the first level includes sensors that collect data such as pressure and water levels, and actuators, which control the state of the system, such as pumps and motors. The second level consists of programmable logic controllers (PLCs), which control and collect information about the system's state, usually storing the data in remote terminal units (RTUs).

The third level is supervisory control, managed by the master terminal unit (MTU), which communicates with the RTUs to send commands and query data. The fourth level is the human-machine interface (HMI), used by operators to view data and control the system through the MTU, with communication facilitated by protocols such as Profibus, Fieldbus, Modbus and DNP3.

III. Methodology

This study was conducted entirely with free software: all experiments were run on a Debian 12 Linux server using the scikit-learn library in Python. Initially, the WUSTL-IIOT-2018 dataset, which contains all the data necessary to perform the experiments, was used. Then, the data was divided into training and testing. From there, the machine learning algorithms were implemented and selected. In the next step, each model was evaluated based on performance metrics, with error and success rates, seeking to identify the efficiency of each algorithm in classifying traffic between attack (1) and benign traffic (0). After evaluating the models, the algorithms were compared to verify which one presented the best performance according to the established metrics, making it possible to answer the work's

motivating question. Figure 1 illustrates the systematic flow of the practical methodology adopted in this research.

The letters (A-H) represent the most relevant steps in the process and can be analyzed in the following sequence:

- Step A: obtaining data that represent network flows for benign and malicious traffic in SCADA networks;
- Step B: beginning of data preprocessing. In this step, the data were discretized and numbered to ensure proportionality and numerical representativeness;
- Step C: continuation of preprocessing: the data are normalized by the "min-max" function to ensure that there are no values < 0 or > 1;
- Step D: completion of preprocessing, where null, duplicate, non-numeric (NaN), or infinite (Inf) values are removed;
- Step E: represents the output of the preprocessing functions. In this step, the data is preprocessed and ready for the training and testing stages;
- Step F: implementation of cross-validation, although more expensive (computationally speaking), is more reliable and stable; Stratified k-fold was used to maintain proportionality between classes in the k-fold rounds. In this step, 10 data folds were created, which allowed 10 rounds of training (90% of the data) and testing (10% of the data) using the 10-fold cross-validation technique;
- Step G: here, several binary supervised classifiers were implemented (Decision Tree, Neural Network, k-Nearest Neighbors, Naive Bayes, Gradient Boosting Machine, and Decision Forest) so that there would be a broad scenario for performance comparison;
- Step H: finally, an intrusion detection model was generated for each classifier (during k-fold cross-validation). The average results of the 10 rounds allowed for extracting True Positive, False Positive, True Negative, and False Negative values, which are essential for analyzing performance through robust metrics such as accuracy, precision, and recall.

A. Dataset

WUSTL-IIOT-2018 is a dataset created and made available as part of a study conducted by Washington University in St. Louis (WUSTL) within the context of cybersecurity research, with a specific focus on SCADA systems. It was built using a test environment designed to emulate real-world industrial systems. Although it doesn't capture seasonal variations (common in SCADA environments), it is quite representative within the proposed context. Its strong imbalance indicates that caution should be exercised in the training strategy chosen in the methodology.

Network traffic is monitored and recorded in a csv format for analysis. It also provides statistics on the captured traffic, indicating the proportion of normal and attack traffic. After preparation, the dataset is classified and labeled to identify normal and attack traffic events. The WUSTL-IIOT-2018 dataset spans 627 MB, is collected over 25 hours, and consists of over 7 million observations. Of these observations, approximately 93.93% correspond to normal network traffic, while the remaining 6.07% represent abnormal traffic resulting from simulated cyberattacks. A closer look reveals that only a tiny fraction of the abnormal traffic is associated with different types of attacks, with 0.0003% for port scanner attacks, 0.0075% for address scanning attacks, 0.0001% for device identification attacks, and 1.1312% for exploit attacks. Notably, the aggressive variation of device identification attacks accounts for about 4.9309% of the abnormal traffic, highlighting its frequency relative to the other attack types.

B. Algorithm Selection and Implementation

1) KNN: KNN (K-Nearest Neighbors) is usually one of the first algorithms to be learned in the area of machine learning. This algorithm analyzes each dataset sample by analyzing its distance from its nearest neighbors. If any of these neighbors are from a class, the selected sample will be classified in that category.

The WUSTL-IIOT-2018 dataset contains several features that describe the traffic of a SCADA network. Each instance of this dataset is labeled as "0" for benign and "1" for attack. When a new instance of network traffic is received, KNN analyzes this instance and calculates its distance from existing instances in the dataset.

2) Neural Network: A neural network is a model inspired by the human brain. It comprises neurons organized into layers: input, hidden, and output. Each neuron receives inputs, performs calculations, and passes the results to the neurons in the next layer. Training involves adjusting the connections, allowing the network to learn from historical data and identify complex patterns.

The goal is to create a model capable of making accurate predictions on new data applicable to image recognition and time series forecasting.

3) Decision Tree: Decision Tree is an algorithm used to solve classification and regression problems, functioning as a decision diagram, where data is separated into groups according to specific characteristics. Its structure resembles an inverted tree, containing a "root node" at the top and "leaves" at the base.

The Decision Tree classifier uses data attributes to perform successive divisions, organizing them hierarchically into internal nodes representing each decision. At each division, metrics

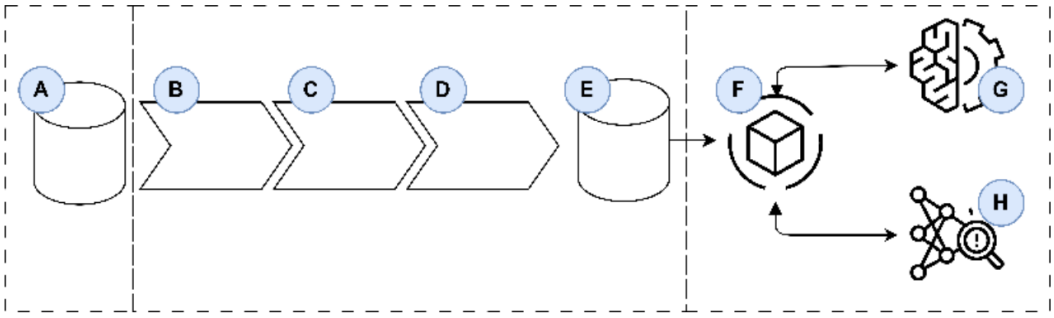

Fig. 1. Proposed methodology workflow.

such as Information Gain or Gini Impurity are implemented to find the best way to separate the data. This process continues until the data reaches the leaf nodes, where the classifications will be assigned.

4) Random Forest: The Random Forest algorithm is composed of several Decision Forests to achieve a single result, which is possible due to a technique called (bagging), where this creates multiple samples of the original dataset to train different decision trees independently. Each tree votes on the class of a sample, and the majority vote determines the final class. In the case of regression, the average of the trees' predictions is used, reducing the risk of overfitting.

5) Naive Bayes: Naive Bayes is an algorithm based on Thomas Bayes' theorem, in which the term "naïve" refers to how the algorithm analyzes the features of a data set, assuming that the features are independent of each other, in addition to assuming that the features variables are of relevant importance to the result.

To perform the prediction calculation, the classifier establishes a probability table containing the predictors' frequencies in relation to the output variables. Finally, the calculation takes into account the highest probability of providing an answer.

6) Gradient Boosting: Gradient Boosting is a machine learning algorithm focused on classification and regression issues, where it combines several weak models (usually Random Forest) to form a stronger and more accurate model. The main idea of this algorithm is to create models sequentially, where each new model seeks to correct the errors of the old models.

The difference between Gradient Boosting and AdaBoost is that GB does not give more weight to items that are classified incorrectly but seek to optimize the loss function so that the current base model is more effective than its predecessor.

C. Confusion Matrix

The confusion matrix is a table designed to facilitate visualization of a classification algorithm's performance. It presents in more detail the classification models' results and compares the model predictions with the real data values.

		Predicted	
		Positive	Negative
Real	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Fig. 2. Confusion Matrix.

- TP - True Positive: corresponds to the number of attack instances classified as an attack.
- FP - False Positive: is the number of benign instances incorrectly classified as attack.
- TN - True Negative: corresponds to the number of benign instances correctly classified as benign.
- FN - False Negative: refers to the number of attack instances incorrectly classified as benign.

D. Model evaluation

To evaluate the performance of the algorithms applied to WUSTL-IIOT-2018, performance metrics were used to measure the effectiveness of the models in identifying normal traffic and attacks.

- AUC (Area Under The Curve): The AUC represents the area under the ROC curve, which measures the model's ability to distinguish between positive and negative classes. The higher the AUC, the better the model's performance in separating classes.

- CA (Acuraccy): Accuracy calculates the proportion of correct predictions about the total samples evaluated:

$$\text{CA} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Precision (P): Precision measures the proportion of true positives about the total number of positive predictions:

$$P = \frac{TP}{TP + FP} \tag{2}$$

- Recall (R): Recall calculates the proportion of true positives about the total number of instances that should have been classified as positive.

$$R = \frac{TP}{TP + FN} \tag{3}$$

- F1-Score: The F1-score is the harmonic mean between Precision and Recall, promoting a balance between these two metrics:

$$\text{F1} = 2 * \frac{P * R}{P + R} \tag{4}$$

- Matthews's Correlation Coefficient (MCC): The MCC measures the correlation between the predictions and the actual values, considering all categories:

$$\text{MCC} = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

IV. Results

Table 3 shows the results obtained according to the performance of the algorithms used to classify network traffic:

Table I
Algorithms's performance comparison.

Model	AUC	ACC	F1-Score	P	R	MCC
Gradient Boost	1.000	1.000	1.000	1.000	1.000	1.000
kNN	1.000	1.000	1.000	1.000	1.000	0.996
Naive Bayes	0.997	0.992	0.993	0.993	0.992	0.936
Neural Network	0.999	0.999	0.999	0.999	0.999	0.987
Random Forest	1.000	1.000	1.000	1.000	1.000	1.000
Decision Tree	0.997	1.000	1.000	1.000	1.000	0.995

All algorithms obtained good results in terms of performance in detecting attacks and classifying benign traffic. Naive Bayes, Random Forest and Decision Tree, although showing a slight drop in the values of AUC, CA, F1, Precision, Recall and MCC, still achieved high indices, close to 1, which demonstrates satisfactory performance, but slightly lower than the other models:

Table II
KNN

	Predicted		
	0	1	$\sum$
Actual	331572	133	331705
	20	20175	20195
	331592	20308	351900

Table III
Decision Tree

	Predicted		
	0	1	$\sum$
Actual	331637	68	331705
	104	20091	20195
	331741	20159	351900

Table IV
Gradient Boosting

	Predicted		
	0	1	$\sum$
Actual	331701	4	331705
	0	20195	20195
	331701	20199	351900

Table V
Random Forest

	Predicted		
	0	1	$\sum$
Actual	331699	6	331705
	1	20194	20195
	331700	20200	351900

Table VI
Rede Neural

	Predicted		
	0	1	$\sum$
Actual	331349	356	331705
	130	20065	20195
	331479	20421	351900

Table VII
Naive Bayes

	Predicted		
	0	1	$\sum$
Actual	329140	2565	331705
	95	20100	20195
	329235	22665	351900

Observing Tables II to VII, it is possible to notice that the results of the confusion matrices demonstrated a comparative analysis between the different classification algorithms used in the experiment. KNN obtained a satisfactory performance in general, with few false positives and negatives. With a lower precision, the Decision Tree classifier slightly presented more errors about KNN. The Gradient Boosting and Random Forest algorithms stood out as being more precise than all the others compared, with almost no classification errors of false positives and false negatives, proposing a considerable capacity to distinguish between benign and malicious traffic. The Neural

Network presented an average performance, registering significantly about the other classifiers. Meanwhile, Naive Bayes presented, among all, the least favorable performance, with a high number of false positives and a relevant number of false negatives.

V. Conclusion and Future Works

This study investigated the potential of machine learning classifier algorithms for intrusion detection in SCADA systems. Based on experiments conducted with the WUSTL-IIOT-2018 dataset already prepared for this purpose, the ability of the algorithms to correctly classify traffic as normal or malicious was analyzed. Using the Orange platform, comparing these classifiers based on performance metrics obtained from confusion matrices was possible. The results demonstrated that machine learning techniques significantly improved intrusion detection and reduced the incidence of false positives and false negatives, contributing to a safer environment. The Gradient Boosting model presented the best performance, with only four false positives and zero false negatives, followed by Random Forest, which had six false positives and one false negative. These numbers demonstrate the effectiveness of these algorithms for the security of SCADA systems. In contrast, Naive Bayes presented a considerably higher number of errors, with 2565 false positives and 95 false negatives, proving to be less suitable for this type of task. The confusion matrices of each algorithm allowed a more detailed visualization of the errors, answering the question about the contribution of each model to the correct classification of attacks. Thus, the analysis reinforces that machine learning models can be valuable tools for protecting these systems when well-adjusted and applied with appropriate algorithms. For future work, this study will likely motivate the exploration of new approaches and algorithms, aiming to increase the detection rate and reduce classification errors, promoting improved security and better adaptation to different industrial contexts.

Acknowledgments

The authors are grateful to State Center for Technological Education “Paula Souza” (CEETEPS) for research conditions at the Defensive Cybersecurity and Artificial Intelligence Laboratory (Detect.AI), and to Research Support Foundation of São Paulo State (FAPESP), Brazil grants #2023/12830-0.

References

[1] R. Lopez Perez, F. Adamsky, R. Soua, and T. Engel, “Forget the myth of the air gap: Machine learning for reliable intrusion detection in scada systems,” *EAI Endorsed Transactions on Security and Safety*, vol. 6, no. 19, jan 2019.

[2] T. Öztürk, Z. Turgut, G. Akgün, and C. Köse, “Machine learning-based intrusion detection for scada systems in healthcare,” *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 11, no. 1, p. 47, 2022.

[3] S. Mokhtari, A. Abbaspour, K. K. Yen, and A. Sargolzaei, “A machine learning approach for anomaly detection in industrial control systems based on measurement data,” *Electronics*, vol. 10, no. 4, p. 407, February 2021.

[4] L. A. C. Ahakonye, C. I. Nwakanma, J. M. Lee, and D. S. Kim, “Scada intrusion detection scheme exploiting the fusion of modified decision tree and chi-square feature selection,” *Internet of Things*, vol. 21, p. 100676, 2023.

[5] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, “Intrusion detection in scada based power grids: Recursive feature elimination model with majority vote ensemble algorithm,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2559–2574, May 2021.

[6] S. Oyucu, O. Polat, M. Türkoğlu, H. Polat, A. Aksöz, and M. T. Ağdaş, “Ensemble learning framework for ddos detection in sdn-based scada systems,” *Sensors*, vol. 24, no. 1, p. 155, Dec 2023.

[7] M. Zaman, D. Upadhyay, and C.-H. Lung, “Validation of a machine learning-based ids design framework using ornl datasets for power system with scada,” *IEEE Access*, vol. 11, pp. 118 414–118 426, Nov 2023.

[8] J. Qian, X. Du, B. Chen, B. Qu, K. Zeng, and J. Liu, “Cyber-physical integrated intrusion detection scheme in scada system of process manufacturing industry,” *IEEE Access*, vol. 8, pp. 147 471–147 481, Aug 2020.

[9] S. Osken, E. Yildirim, G. Karatas, and L. Cuhaci, “Intrusion detection systems with deep learning: A systematic mapping study,” in *2019 Scientific Meeting on Electrical-Electronics Biomedical Engineering and Computer Science (EBBT)*, 2019, pp. 1–4.

[10] J. Anderson, “Computer security threat monitoring and surveillance,” James P. Anderson Company, Tech. Rep., 1980.

[11] D. Denning, “An intrusion-detection model,” *IEEE Transactions on Software Engineering*, pp. 222–232, 1987.

[12] G. Karatas and O. Sahingoz, “Neural network based intrusion detection systems with different training functions,” in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, 2018, pp. 1–6.

[13] M. Kaouk, J. Flaus, M. Potet, and R. Groz, “A review of intrusion detection systems for industrial control systems,” in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019, pp. 1699–1704.

[14] J. Ran, Y. Ji, and B. Tang, “A semi-supervised learning approach to ieee 802.11 network anomaly detection,” in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 2019, pp. 1–5.

[15] S. Alam, M. Shuaib, and A. Samad, “A collaborative study of intrusion detection and prevention techniques in cloud computing,” in *International Conference on Innovative Computing and Communications*, 2019, pp. 231–240.

[16] I. P. Turnipseed, “A new scada dataset for intrusion detection system research,” Master of Science Thesis, Mississippi State University, Starkville, Mississippi, USA, August 2015, datasets include network traffic captured on a gas pipeline SCADA system in MSU’s SCADA lab. [Online]. Available: https://scholarsjunction.msstate.edu/td/209/

[17] M. S. Altaha and H. Hong, “Anomaly detection for scada system security based on unsupervised learning and function codes analysis in the dnp3 protocol,” 2022.