Securing SCADA Systems: A Comprehensive Machine Learning Approach for Detecting Reconnaissance Attacks

Ezaz Aldahasi[1] and Talal Alkharobi[2]

[1]Information & Computer Science Department, KFUPM, Saudi Arabia
[2]Computer Engineering Department, KFUPM, Saudi Arabia

Abstract

Ensuring the security of Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems (ICS) is paramount to safeguarding the reliability and safety of critical infrastructure. This paper addresses the significant threat posed by reconnaissance attacks on SCADA/ICS networks and presents an innovative methodology for enhancing their protection. The proposed approach strategically employs imbalance dataset handling techniques, ensemble methods, and feature engineering to enhance the resilience of SCADA/ICS systems. Experimentation and analysis demonstrate the compelling efficacy of our strategy, as evidenced by excellent model performance characterized by good precision, recall, and a commendably low false negative (FN). The practical utility of our approach is underscored through the evaluation of real-world SCADA/ICS datasets, showcasing superior performance compared to existing methods in a comparative analysis. Moreover, the integration of feature augmentation is revealed to significantly enhance detection capabilities. This research contributes to advancing the security posture of SCADA/ICS environments, addressing a critical imperative in the face of evolving cyber threats.

Keywords:
SCADA/ICS security; Reconnaissance attacks; Critical infrastructure; Imbalanced dataset handling techniques; Ensemble methods; Feature Engineering

1. Introduction

SCADA/ICS systems are critical infrastructure systems that require robust security measures to protect against potential cyber threats. There are security measures that help to protect these systems from potential cyber threats and ensure the integrity, availability, and confidentiality of critical infrastructure operations [1]. Here are some key considerations for SCADA system security [2]:

- Network Segmentation
- Access Control
- Encryption
- Intrusion Detection and Prevention Systems (IDPS)
- Patch Management
- Security Monitoring and Logging
- Security Awareness and Training

Malicious actors are showing a growing interest in targeting SCADA/ICS systems to exploit vulnerabilities and cause disruption to critical infrastructure operations [3]. Since SCADA systems typically monitor and control critical infrastructures, they are targeted by technically skilled and well-organized attackers, called adaptive persistent adversaries. There is a lack of research examining attacks targeting operational technology (OT) in a SCADA/ICS system, such as attacks on the PLC protocol [4] One of these attacks is reconnaissance attacks which is essential in the attacker's strategy since they collect vital information regarding the vulnerabilities and architecture of the target system. It is a serious threat to the infrastructure of SCADA systems. The security of these systems is crucial, not only for safeguarding against hostile and cyber-terrorist attacks but also for ensuring the resilience and integrity of processes and activities, given their indispensable role in the economy [5]. This is why this is a current field of research where concrete improved solutions to SCADA/ICS systems security are anticipated. Our system looks for the packets affected by reconnaissance attacks which are launched through the attacker's machine against the simulator of the SCADA/ICS system. Once the network traffic is captured, the next step is to select potential features

Manuscript received December 5, 2023
Manuscript revised December 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.12.1

that can distinguish the anomalous traffic from the normal traffic. Features during the normal and attack traffic will be analyzed, as well as those features that did not vary during the normal and attack traffic. To sum up, this work suggests a complete framework that integrates many strategies, including unbalanced dataset handling, ensemble methods, and feature augmentation, to strengthen the defense mechanisms against these attacks.

2. Literature Review

This section presents the prior research on the topics related to SCADA/ICS system security in two sections as detailed below.

2.1 Methods and Techniques for Protecting SCADA Systems for Cyber Security

Due to the importance of network security in SCADA systems. The authors in [6] proposed the use of network reconnaissance and firewall filters to enhance SCADA network security. The researchers investigated the notion of reconnaissance assaults, the methodologies employed in active and passive reconnaissance, and the efficacy of firewall filters in thwarting potentially malicious network data. They presented an experiment using a Python port scanner to perform network reconnaissance on SCADA systems. In addition, they pointed to the different techniques used in active and passive reconnaissance attacks. Several techniques employed in active reconnaissance assaults include host sweep attacks; port scan attacks; and service scan attacks. In contrast, passive network reconnaissance involves the collection of network information without engaging in direct examination of the target network. Some techniques used in passive reconnaissance attacks include network traffic monitoring and intrusion detection systems. Those techniques serve distinct functions in conducting reconnaissance operations and can be employed by anyone with malicious intent to collect data pertaining to the weaknesses of a certain network target. The outcome of this study demonstrates that network reconnaissance methodologies, such as port scanning, can be effectively employed to acquire data pertaining to SCADA systems. Nevertheless, the implementation of firewall filters can successfully impede dubious network traffic and bolster the security of SCADA networks by thwarting illegal entry and mitigating potential vulnerabilities. The trials carried out in the study provide evidence of the efficacy of these strategies in enhancing the security of SCADA networks. Machine learning algorithms have been used to detect attacks cyber-attacks in SCADA power systems [7], where the authors compared the performance of individual machine learning models and demonstrated that ensemble methods outperform them in terms of accuracy, false alarm rate, and non-detection rate. They concluded the importance of ensemble learning in improving prediction and detection accuracy in industrial control systems. They analyzed and compared 10 well-known traditional machine learning algorithms for cyber-attack detection in SCADA systems. These algorithms include Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Stochastic Gradient Descent (SGD), Gradient Boosting Classifier (GBC), Light Gradient Boosting Machine (LGBM), and eXtreme Gradient Boosting (XGB). The paper also utilizes stacking ensemble learners with different meta-learners to improve prediction performance. The specific tools or software used for implementing these algorithms and ensemble methods are not mentioned in this work. The use of adversarial machine learning attacks on supervised machine learning classifiers in Industrial Control Systems (ICS) is explored [8].The study evaluates the performance of different classifiers, such as Random Forest and J48, on original and adversarial samples generated using the Jacobian-based Saliency Map Attack (JSMA). The authors discussed the effectiveness of adversarial training as a defense mechanism and suggested further research on generating adversarial samples and exploring other defense mechanisms. They highlighted the potential risks and challenges associated with using machine learning-based intrusion detection systems in SCADA networks. This understanding is crucial for developing more robust and effective security mechanisms for protecting SCADA systems against cyber threats. The application of machine learning algorithms in various SCADA datasets includes attack detection and classification in Industrial Control Systems. These algorithms are used to analyze the data collected from SCADA systems and identify potential cyber-attacks or anomalies. Some of the machine learning algorithms commonly applied in SCADA datasets include Naive Bayes, Random Forest, SVM, J48, and Neural Networks. For example, in a study by Yaghoubi and Fainekos [8] a gradient-based search approach was used to evaluate the effectiveness of machine learning algorithms, specifically Recurrent Neural Networks (RNN), in detecting attacks in a Simulink model of a steam condenser. Erba et al. demonstrated real-time evasion attacks using RNN models and utilized an autoencoder to generate adversarial samples [9]. Furthermore, machine learning algorithms have been applied in various SCADA datasets such as gas pipelines, power systems, wind turbines, and SCADA testbeds. These algorithms are used to train classifiers that can detect and classify different types of attacks or abnormal behavior in SCADA systems. Overall, the application of machine learning algorithms in SCADA datasets enables the development of robust intrusion detection systems that can effectively detect and mitigate cyber threats in Industrial Control Systems.

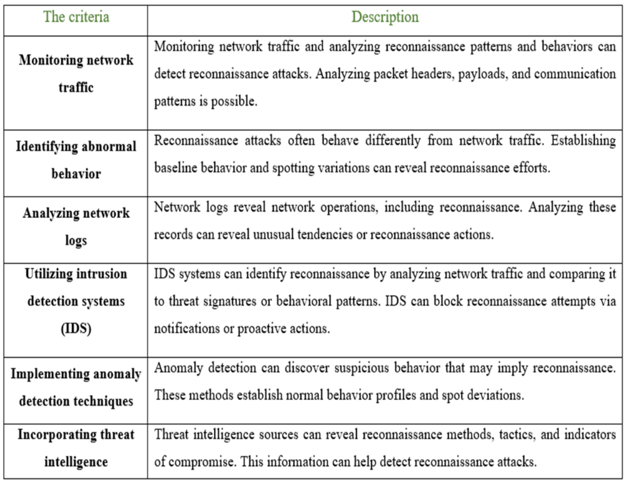
2.2 Techniques of Detection of Reconnaissance Attacks

Authors in [10] proposed a defense mechanism called DefRec to protect power grids from reconnaissance attacks. They used physical function virtualization (PFV) to create lightweight virtual nodes that mimic the behavior of real devices, making it difficult for adversaries to identify them. Their framework includes security policies to randomize network communications and craft decoy data, disrupting adversaries' reconnaissance efforts. The evaluation shows that PFV accurately follows the behavior of real devices and DefRec significantly delays adversaries' reconnaissance efforts. Overall, their work provides effective protection against reconnaissance attacks, intelligent adversaries, and SDN vulnerabilities, while considering practical threat scenarios and minimizing performance impact. In [11], authors proposed a technique called Random Host Address Mutation (RHM) to disrupt reconnaissance attacks in computer networks. It randomizes attributes of network hosts, including their IP address, MAC address, and domain name. They focused on IP address mutation, which is the most effective randomization vector against reconnaissance attacks. They discussed the implementation and effectiveness of RHM in both legacy networks and Software-Defined Networks (SDN). RHM disrupts reconnaissance attacks in computer networks by deprecating the adversary's information about the network, forcing the attacker to frequently redo their reconnaissance activities to regain the lost information, thus delaying the completion of the attack. This disruption is achieved through fast address randomization, which changes the addresses of network hosts and invalidates existing mappings, forcing potential attackers to waste their resources on the re-discovery of these mappings. RHM achieves high uncertainty in adversary scanning by modeling address mutation randomization as a multi-level satisfiability problem, allowing for highly unpredictable and fast address randomization. Additionally, RHM separates mutation from end-hosts and manages it via network appliances, enabling a high mutation rate and making it difficult for attackers to keep up with the changing network addresses. Some authors proposed a lightweight algorithm for detecting and blocking reconnaissance attacks and discussed the vulnerability of IoT devices to such attacks [12]. They conducted experiments on a Raspberry Pi host and found that changes in packet size and count can be used to detect and block reconnaissance attacks. They developed a Python program for real-time packet capture and implemented an automated scan detection and blocking process.

3. Requirements of Reconnaissance Attack Detection in Systems

The criteria utilized for the detection of reconnaissance assaults in systems have been defined and derived from prior research. These criteria are outlined as shown in Table.1

Table.1 Requirements of Reconnaissance Attack Detection in Systems [13]



The criteria	Description
Monitoring network traffic	Monitoring network traffic and analyzing reconnaissance patterns and behaviors can detect reconnaissance attacks. Analyzing packet headers, payloads, and communication patterns is possible.
Identifying abnormal behavior	Reconnaissance attacks often behave differently from network traffic. Establishing baseline behavior and spotting variations can reveal reconnaissance efforts.
Analyzing network logs	Network logs reveal network operations, including reconnaissance. Analyzing these records can reveal unusual tendencies or reconnaissance actions.
Utilizing intrusion detection systems (IDS)	IDS systems can identify reconnaissance by analyzing network traffic and comparing it to threat signatures or behavioral patterns. IDS can block reconnaissance attempts via notifications or proactive actions.
Implementing anomaly detection techniques	Anomaly detection can discover suspicious behavior that may imply reconnaissance. These methods establish normal behavior profiles and spot deviations.
Incorporating threat intelligence	Threat intelligence sources can reveal reconnaissance methods, tactics, and indicators of compromise. This information can help detect reconnaissance attacks.

4. The patterns and Behaviors of Reconnaissance Activities

A set of features were also deduced by which reconnaissance attacks were detected and distinguished where the detection of reconnaissance attacks relies on the identification of many key aspects inside network traffic. These features include [14]:

High volume of connection attempts: Reconnaissance attacks often involve scanning many IP addresses or ports. Monitoring a high volume of connection attempts from a single source IP address or a series of sequential port numbers can indicate a reconnaissance activity.

Unusual scanning patterns: Reconnaissance activities may exhibit specific scanning patterns, such as sequential scanning, random scanning, or scanning specific ranges of IP addresses. Analyzing the packet headers and payload contents for these scanning patterns can help detect reconnaissance attacks.

Information gathering: Adversaries may collect information about the target system, such as IP addresses, domain names, email addresses, or user accounts. This can be detected by monitoring unusual data requests or information-gathering activities.

Unusual ICMP traffic: ICMP (Internet Control Message Protocol) packets can be used for reconnaissance purposes, such as ICMP Echo Requests (ping) or ICMP Timestamp Requests. Monitoring for unusual or excessive ICMP traffic can help detect reconnaissance activities.

Rapid changes in the windowed averages of packet size and packet count: The authors found that during a reconnaissance attack, there are rapid changes in the magnitude of these performance metrics. These alterations can be detected with minimal computational effort and serve as indicators of a reconnaissance attack.

Unauthorized access attempts: Adversaries may attempt to gain unauthorized access to systems or devices by trying default or weak credentials, exploiting known vulnerabilities, or conducting brute-force attacks. Unusual login attempts or authentication failures can indicate reconnaissance activities. It is essential to note that these patterns and behaviors may vary depending on the context and character of the reconnaissance activities.

5. Methodology

This section elucidates the technique utilized in this work. Data preprocessing has been completed, resulting in the creation of a classifier model. The efficacy of the model is evaluated by performance assessment. The following are the prescribed procedures for reaching findings.

A. Dataset Description

The UNSW-NB15 dataset is a comprehensive collection of network intrusion data [15]. It includes a wide range of variables that are associated with network traffic, as well as labels that indicate whether each instance is classified as normal or reflects a specific form of network intrusion. The dataset comprises 49 features, along with the class label, which are grouped into sets as follows:

Packet-Level Features encompass many details derived from network packets, including source and destination IP addresses, source and destination ports, protocol type (TCP, UDP, etc.), packet size, time-to-live (TTL), and other relevant information.

Flow-Level Features over a series of packets, aggregations, and statistical measures are computed, creating a "flow." In essence, flows are the relationships that exist between a source and a destination during a given duration. At this level, features can include the number of bytes transported, the length of the flow, and the number of packets in the flow.

Statistical Features encompass a range of measurements, including mean, standard deviation, minimum, maximum, and others, which are computed across the packet or flow-level characteristics.

Class Label the label denotes whether a given occurrence represents regular network activity or a distinct form of network intrusion. The classification is typically binary, distinguishing between normal and intrusion instances. However, the dataset may have numerous classes to represent different types of intrusions.

The original dataset consists of 2,540,044 instances and nine categories of attacks: Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms. However, for our analysis, we focus just on the target instances, specifically reconnaissance and regular packets, which amount to 50,290 instances. The tools and services, including a variety of protocols that can be associated with network communication to launch reconnaissance attacks are presented in Table 2. It is common practice for reconnaissance attacks to involve the utilization of a variety of tools and protocols to acquire a full understanding of the topology, services, and potential vulnerabilities of the target network.

Telnet: Telnet might be used by attackers to probe a network and locate weak services or open ports where they could exploit vulnerabilities.
SNMP (Simple Network Management Protocol): Utilizing the Specialized Network Management Protocol (SNMP) allows for the collection of data regarding the configurations, devices, and performance of a network.
SunRPC Portmapper (TCP/UDP): Service Enumeration involves the identification of services that are registered with the SunRPC Portmapper to gain a comprehensive understanding of the available network services.
NetBIOS: Network enumeration involves gathering data regarding network shares, users, and system specifics within Windows environments.
DNS (Domain Name System): DNS reconnaissance involves gathering data on a network's domain architecture and the corresponding IP addresses.
HTTP: Web Application Reconnaissance is the examination of web servers to identify vulnerabilities or misconfigurations.
ICMP (Internet Control Message Protocol): is a network layer protocol that is part of the Internet Protocol (IP) suite. It is used for diagnostic and control purposes in networking. ICMP messages are typically generated by network devices, such as routers or hosts, to communicate error conditions or provide other information about the network.
Ping Sweeps: Detecting active hosts within a network.
SCTP (Stream Control Transmission Protocol): Network scanning is the process of identifying hosts and detecting open ports.
MSSQL: Database Reconnaissance involves the identification of Microsoft SQL Server instances and the potential exploitation of vulnerabilities.
SMTP (Simple Mail Transfer Protocol): Email Reconnaissance involves the process of collecting data from email servers and addresses.

Figure 1. shows the distribution of dataset based on our target feature (attack_cat) and Table 3 represents the class counts based on Attack_cat feature.

Table 2. Attack and Types [15]

Attack Category	Tools and Services
Reconnaissance	Telnet
	SNMP
	SunRPC Portmapper (TCP) UDP Service
	SunRPC Portmapper (TCP) TCP Service
	SunRPC Portmapper (UDP) UDP Service
	NetBIOS
	DNS
	HTTP
	SunRPC Portmapper (UDP)
	ICMP
	SCTP
	MSSQL
	SMTP

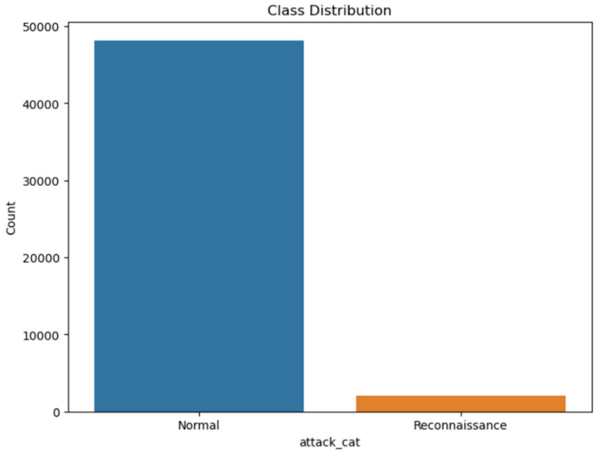


Figure.1 Class Distribution Based on the Target Feature

Table 3. Class Counts based on Attack_cat Feature

Attack category	Count
Normal	48205
Reconnaissance	2085

B. Data preprocessing

- **Handling Missing Values**

In this step, the dropna() function is employed to identify and remove any missing values. This method is frequently utilized in data analysis and manipulation libraries like pandas in Python. This function is employed to eliminate any missing or null values included in a dataset.

- **Feature Engineering**

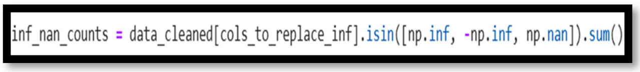
By adding new features, we aimed to increase the performance and quality of the models or algorithms used for classification or prediction tasks, as well as the detection of reconnaissance assaults.

Using the features of the current dataset, Table 4 represents the extra features that were computed.

Table 4. Information of New Features

Feature	Information
packet_ratio	**Definition**
	The proportion of source to destination packets (spkts to dpkts).
	Calculation
	data['packet_ratio'] = data['spkts'] / data['dpkts']
bytes_ratio	**Definition**
	The ratio of source bytes (sbytes) to destination bytes (dbytes).
	Calculation
	data['bytes_ratio'] = data['sbytes'] / data['dbytes']
time_ratio	**Definition**
	The ratio of the live value (sttl / dttl) of the source and destination times.
	Calculation
	data['time_ratio'] = data['sttl'] / data['dttl']
retransmission_rate	**Definition**
	The rate of dropped or retransmitted packets; it is computed by dividing the total number of packets (spkts + dpkts) by the sum of destination and source dropped or retransmitted packets (dloss).
	Calculation
	data ['retransmission_rate'] = (data['sloss'] + data['dloss']) / (data['spkts'] + data['dpkts'])
common_service	**Definition**
	indicates whether or not the service type (service) is a common service (such as HTTP, FTP, or SMTP).
	Calculation
	data['common_service'] = data['service'].apply(lambda x: 1 if x in ['HTTP', 'FTP', 'SMTP'] else 0)

Following the addition of those features, the following command was used to investigate the unique values in each column and determine whether there are infinite values or NaN.



```
inf_nan_counts = data_cleaned[cols_to_replace_inf].isin([np.inf, -np.inf, np.nan]).sum()
```

- **Handling Categorical Variable**

One-hot encoding is a method employed in machine learning and data preprocessing to convert category variables into binary vectors. It is especially advantageous when working with categorical data in a manner that machine learning models can successfully handle. We employ one-hot encoding on the 'proto', 'service', and 'state' columns in our work. By using one-hot encoding, the data is converted into a binary matrix, where each column represents a distinct type of protocol, service, and state.

- **Feature selection**

Within this subsection, the appropriate variables are selected from a given dataset. Feature selection is essential since it can improve the performance and accuracy of the approaches. Feature selection is a technique that helps extract crucial information from a large dataset to reduce processing time [16]. After considering the correlation among the features, the following features ('ackdat', 'ct_ftp_cmd', 'dbytes', 'synack', 'state__FIN', 'is_sm_ips_ports', 'dloss', 'dpkts', 'dwin') were removed due to their Pearson correlation coefficient exceeding the threshold limit of 0.9. Figure 2. displays the correlation matrix.

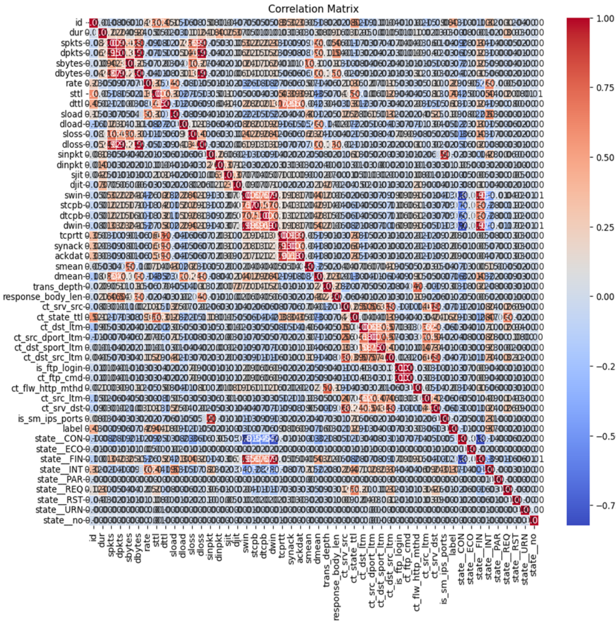


Figure.2 Correlation matrix

- **Imbalance dataset handling**

The inherent class imbalance in SCADA/ICS datasets is addressed by using imbalance dataset handling approaches. Machine learning algorithms tend to favor the dominant class, which makes it difficult for them to identify attacks coming from the minority class.

To tackle this concern, a range of sampling methodologies are implemented, encompassing oversampling, undersampling, and hybrid approaches. Undersampling entails a reduction in the number of instances of the majority class, whereas oversampling increases the number of instances of the minority class. By balancing the dataset, combining oversampling and undersampling, either independently or via sophisticated techniques such as SMOTE (Synthetic Minority Oversampling Technique) with Tomek connections, the objective is to reduce class imbalance.

We used the undersampling approach on the majority class since our dataset is unbalanced as shown in Table 3. The balanced target labeling that distinguishes between "Noraml" and "Reconnaissance" is visually represented in Figure 3.

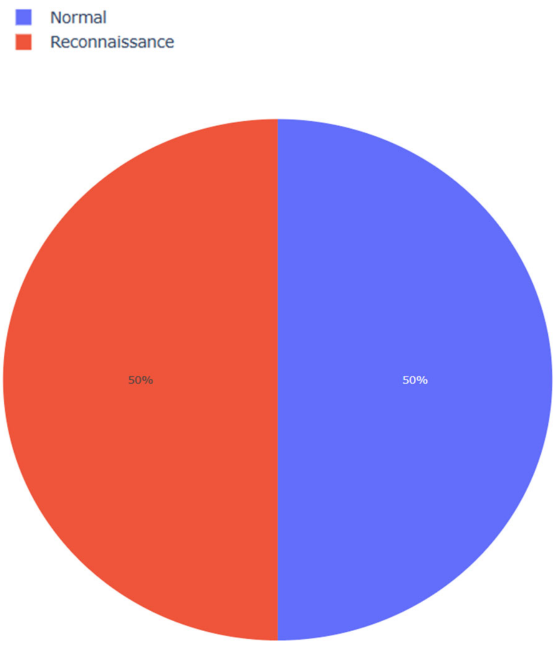


Figure 3. Balance Class Distribution

- **Standardization**

Numerical characteristics are rescaled to have zero mean and unit variance through the preprocessing step of standardization. Each feature's mean is subtracted, and the result is divided by the standard deviation. When characteristics in the dataset have varying scales, it is crucial. Standardization is used in this context to change the numerical properties. This is especially crucial when utilizing machine learning techniques that depend on gradients or measures based on distance.

C. Making the model

Ensemble approaches, renowned for their capacity to enhance prediction accuracy, are employed to further augment the detection of reconnaissance attacks in SCADA/ICS systems. A robust and dependable ensemble model is created by integrating multiple learning methods. The ensemble model utilizes the varied viewpoints of distinct models, leading to improved accuracy and robustness in detecting attacks. The study used Principal Component Analysis (PCA) as a method to reduce the dimensions of the dataset, aiming to capture 95% of the variation included in the original data. The data is divided between training and testing sets using a split ratio of 70-30. A Random Forest classifier is initialized and trained using the training data, and the relevance of its features is calculated. To improve the model's performance, the features are rearranged according to their significance, and only the most important features are kept. Afterwards, the adjusted dataset is used to train a Random Forest classifier, Logistic Regression, K-Nearest Neighbors, Support Vector Machine, and XGBoost Classifier separately. A Voting Classifier is employed to implement an ensemble approach, which combines the predictions made by the different models.

D. Result and Discussion

When working on identifying abnormal traffic, particularly in the context of detecting reconnaissance assaults, it is crucial to prioritize metrics that offer valuable insights into the accuracy of our model in differentiating between normal and abnormal patterns. The performance of each classifier was assessed using metrics such as Accuracy, Precision, Recall, F1-Score.

Precision: Regarding the identification of reconnaissance attacks, high precision signifies that when the model predicts abnormal traffic, it is very likely to be accurate.

Recall (Sensitivity or True Positive Rate): Recall quantifies the model's capacity to accurately detect and include all occurrences of reconnaissance attacks. A high recall indicates that the model is capable of accurately detecting a substantial proportion of the true abnormal cases.

F1-Score: The F1-score is calculated as the harmonic mean of precision and recall. It offers an equitable trade-off between precision and recall. It is especially beneficial when there is a disparity between normal and reconnaissance attack occurrences.

False Positive (FP): A false positive occurs when the model predicts the positive class (reconnaissance attack) but the true class is negative (normal). In other words, the model incorrectly identifies an instance as belonging to the positive class when it does not.

False Positive (FP) = Number of instances wrongly predicted as positive.

False Positive (FP)=Number of instances wrongly predicted as positive

False Negative (FN): A false negative occurs when the model predicts the negative class (normal) but the true class is positive (reconnaissance attack). In other words, the model fails to identify an instance that actually belongs to the positive class.

False Negative (FN)= Number of instances wrongly predicted as negative

False Negative (FN)=Number of instances wrongly predicted as negative

In the context of detecting attacks, reducing False Negatives is often crucial because it means improving the model's ability to correctly identify instances of attacks. However, this might come at the cost of increasing False Positives. In security applications, minimizing false negatives is often crucial to prevent actual attacks from going undetected. Therefore, prioritizing sensitivity (minimizing FN) might be more important in these scenarios. In addition, ensuring a balance between high precision and memory is vital for identifying reconnaissance attacks. It is frequently deemed acceptable to tolerate a greater incidence of false positives to prevent the possibility of overlooking potential reconnaissance actions. Table 5 displays the findings of the used models based on the most effective performance assessment outcomes on the utilized dataset using the optimal approach of undersampling.

Table 5. Performance Evaluation Results without Threshold

Model	Precision	Recall	F1-score	FP	FN
Random Fo rest	0.98	0.90	0.94	11	65
Logistic Re gression	0.89	0.88	0.88	72	79
K-Nearest Neighbors	0.94	0.89	0.91	34	73
SVC	0.95	0.91	0.93	28	56
XGBoost	0.97	0.93	0.95	19	47

The presented results highlight the performance metrics of various machine learning models in the context of a binary classification task, specifically aimed at detecting instances of attacks. Precision, recall, and F1-score are commonly utilized metrics to evaluate the efficacy of classification models.
Precision: The Random Forest and XGBoost models exhibit high precision values of 0.98 and 0.97, respectively, suggesting a strong ability to correctly identify attack instances.
Recall: The Random Forest and XGBoost models demonstrate respectable recall values of 0.90 and 0.93, respectively, indicating their effectiveness in identifying a substantial portion of actual attack instances.
F1-score: The Random Forest, XGBoost, and SVC models showcase F1-scores of 0.94, 0.95, and 0.93, respectively, suggesting a harmonious trade-off between precision and recall.

It is imperative to consider the occurrences of false positives (FP) and false negatives (FN) in the evaluation. False positives (instances incorrectly predicted as attacks) and false negatives (actual attacks overlooked by the model) are consequential factors. The Logistic Regression model presents a relatively higher count of both FP (72) and FN (79), indicating a potential area for improvement in achieving a more balanced classification. The Receiver Operating Characteristic (ROC) curve provides a comprehensive evaluation of the model's performance across various thresholds. ROC curve for our work is displayed from Figure 4 to Figure 8.

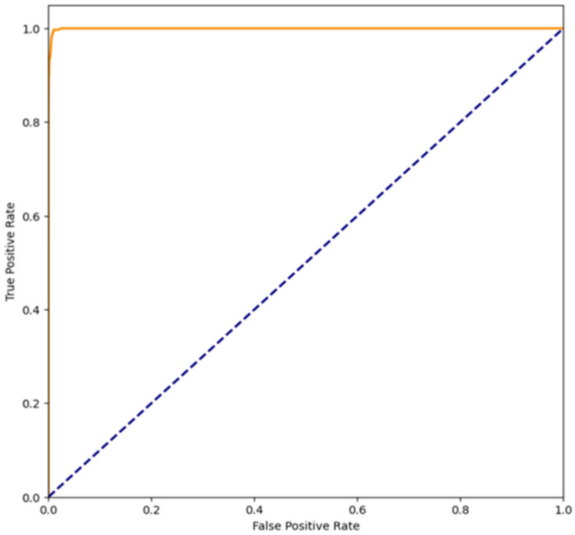


Figure 4. Roc Curve for Random Forest

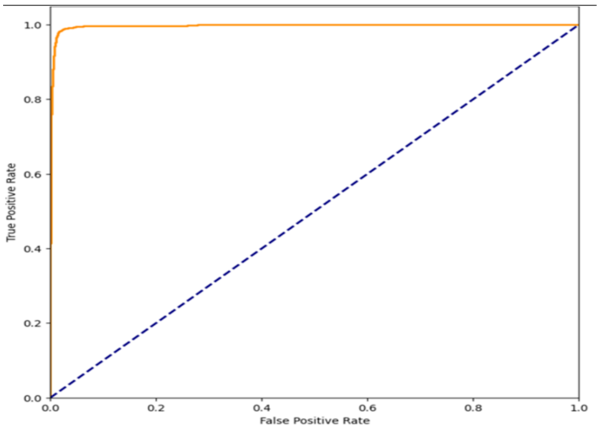


Figure 5. Roc Curve for Logistic Regression

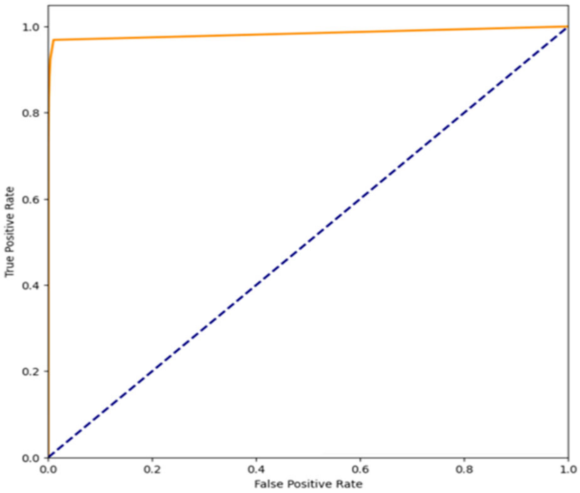


Figure 6. Roc Curve for K-Nearest Neighbors

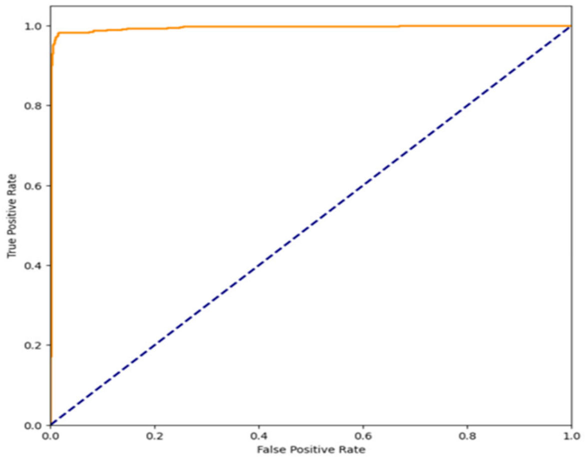


Figure 7. Roc Curve for Support Vector Machine

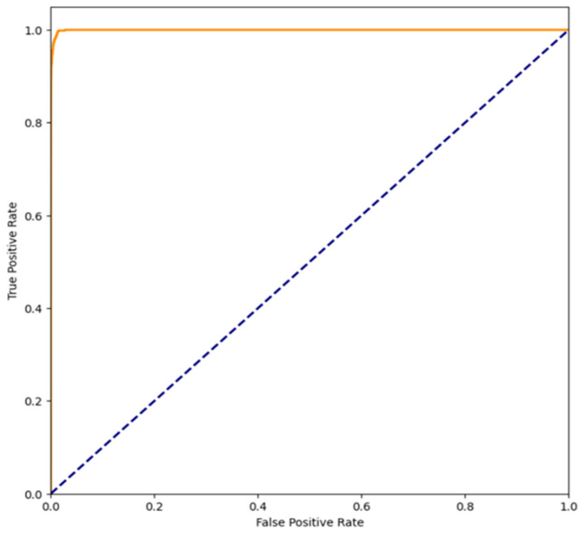


Figure 8. Roc Curve for XGBoost

Notably, we observed a higher FN compared to the FP, which suggests that the model is more prone to missing actual positive cases (occurrences of reconnaissance attacks) while wrongly classifying some negative cases (normal traffic) as positive, which can have serious consequences for the security of the system. To address this issue we applied another strategy, threshold adjustment. Experiment with adjusting the classification threshold. By default, classifiers use a threshold of 0.5 for binary classification. To reduce false negatives, we considered lowering the threshold (to 0.3 for each classifier individually) and used the predict_proba method to get class probabilities and then applied a custom threshold.

Table 6. Performance Evaluation Results with 0.3 Threshold

Model	Precision	Recall	F1-score	FP	FN
Random Forest	0.91	0.95	0.93	63	30
Logistic Regression	0.84	0.93	0.88	113	43
K-Nearest Neighbors	0.91	0.93	0.92	48	59
SVC	0.92	0.93	0.93	49	44
XGBoost	0.95	0.94	0.95	37	33

The reported evaluation metrics of machine learning models in the presented results provide valuable insights into their performance in a binary classification task, with a focus on detecting instances of attacks. Precision, recall, and F1-score are utilized as standard measures to gauge the effectiveness of the models.

Precision: The XGBoost model exhibits the highest precision at 0.95, indicating a strong capability to correctly identify instances of attacks. Random Forest, K-Nearest Neighbors, and Support Vector Classifier (SVC) also demonstrate commendable precision values of 0.91, 0.91, and 0.92, respectively.

Recall: The Logistic Regression model displays the highest recall at 0.93, closely followed by Random Forest, K-Nearest Neighbors, SVC, and XGBoost with recall values of 0.95, 0.93, 0.93, and 0.94, respectively.

F1-score: The XGBoost demonstrates the highest F1-score at 0.95, suggesting a favorable balance between precision and recall. Random Forest, K-Nearest Neighbors, and SVC also present strong F1-scores of 0.93, 0.92, and 0.93, respectively.

The presented results in Table. 6 show better performance in FP and FN counts across models. Nevertheless, Logistic Regression has relatively higher counts of both FP (113) and FN (43), suggesting potential areas for improvement in achieving more. Comparing the performance of different models helps identify which algorithm is better suited for the specific problem. Ensemble methods like Random Forest and XGBoost often provide robust performance.

- **Comparison of our Proposed Approach with Other Work**

In [17], the authors evaluated the performance of several machine learning algorithms in detecting cyber-attacks on a SCADA system testbed. The dataset used in the study is unbalanced, meaning that the distribution of the classes (normal traffic and attack traffic) is uneven. The authors mentioned that the unbalanced dataset affected the evaluation of the algorithms and emphasized the need for other metrics to compare the performance of the machine learning algorithms. They also discussed that the models are biased toward normal traffic due to the unbalanced dataset, and this bias could affect the evaluation metrics such as false alarm rate and undetected rate. They used the FAR (False Alarm Rate) and UND (Un-Detection Rate) metrics to evaluate the performance of the algorithms and founds that the Decision Tree, Random Forest, and KNN models had the lowest FAR values, indicating that they were good at detecting normal traffic. On the other hand, the Naïve Bayes, Logistic Regression, and KNN models had the lowest UND values, indicating that they were good at detecting anomalous traffic. But in general, the results of FPR were greater than FNR in all the traditional ML algorithms used. The unbalanced nature of the dataset is an important consideration in the evaluation of the machine learning algorithms and their performance in detecting attacks. It is essential to address the challenges posed by unbalanced datasets to obtain accurate and reliable results in cybersecurity research.

The usefulness of the suggested approach in detecting reconnaissance assaults is demonstrated through the evaluation of real-world SCADA/ICS datasets and experimental outcomes. The suggested ensemble model outperforms existing methods in a comparative analysis. Additionally, the use of feature augmentation significantly improves detection and reduces false positives. Excellent model performance is indicated by high precision and recall, together with a low FP and FN. Overall, these measures provide a thorough perspective on the performance of the models.

Conclusion

In conclusion, our research represents a significant advancement in bolstering the security of SCADA/ICS systems through the development of a powerful predictive model tailored for detecting reconnaissance attacks. By addressing the inherent challenges associated with class imbalance using state-of-the-art dataset-balancing techniques, we have ensured the model's ability to discern between normal traffic and potential threats.

Our approach incorporates ensemble methods, leveraging the strengths of Random Forest and XGBoost models, both demonstrating high precision (0.98 and 0.97, respectively) and commendable recall (0.90 and 0.93, respectively). This affirms the model's proficiency in accurately identifying instances of attacks while minimizing false negatives.

Notably, our findings shed light on the importance of a well-balanced dataset, where precision, recall, and F1-score serve as key performance indicators. The observed higher count of false negatives in the Logistic Regression model prompted us to explore threshold adjustment as a strategic mitigation measure. By experimenting with lower classification thresholds, we aimed to mitigate the risk of overlooking actual positive cases, especially in the context of reconnaissance attacks, while carefully managing false positives. Furthermore, our research underscores the effectiveness of ensemble methods and the strategic application of innovative features. The resulting model not only fortifies the overall security posture of SCADA/ICS systems but also signifies a pivotal stride in safeguarding critical infrastructure from potential threats. In essence, our work highlights the significance of a holistic and innovative approach to predictive modeling in the realm of SCADA/ICS security. As we navigate the evolving landscape of cybersecurity, our findings contribute valuable insights, paving the way for more resilient and proactive reconnaissance attack detection mechanisms in critical infrastructure environments.

References

[1] A. Shahzad, S. Musa, A. Aborujilah, and M. Irfan, “The SCADA review: System components, architecture, protocols and future security trends,” *Am. J. Appl. Sci.*, vol. 11, no. 8, pp. 1418–1425, 2014, doi: 10.3844/ajassp.2014.1418.1425.

[2] S. Min Han, C. Lee, Y. Ho Chae, and P. Hyun Seong, “A study on classification of the security controls for the effective implementation to nuclear power plant,” *Nucl. Eng. Technol.*, vol. 54, no. 4, pp. 1245–1252, 2022, doi: 10.1016/j.net.2021.10.009.

[3] S. Kim, G. Heo, E. Zio, J. Shin, and J. gu Song, “Cyber attack taxonomy for digital environment in nuclear power plants,” *Nucl. Eng. Technol.*, vol. 52, no. 5, pp. 995–1001, 2020, doi: 10.1016/j.net.2019.11.001.

[4] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, “A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020, doi: 10.1109/COMST.2020.2987688.

[5] F. A. Alhaidari and E. M. Al-Dahasi, “New approach to determine DDoS attack patterns on SCADA system using machine learning,” *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, pp. 1–6, 2019, doi: 10.1109/ICCISci.2019.8716432.

[6] P. Anggraeni, A. R. Harits M., and M. Fikri Radhea, “Identifying SCADA Network Security Through Network Reconnaissance and Firewall Filter,” *ISMEE 2021 - 2021 3rd Int. Symp. Mater. Electr. Eng. Conf. Enhancing Res. Qual. F. Mater. Electr. Eng. a Better Life*, pp. 211–216, 2021, doi: 10.1109/ISMEE54273.2021.9774069.

[7] M. Timken, O. Gungor, T. Rosing, and B. Aksanli, “Analysis of Machine Learning Algorithms for Cyber Attack Detection in SCADA Power Systems,” *2023 Int. Conf. Smart Appl. Commun. Networking, SmartNets 2023*, no. Ml, pp. 1–6, 2023, doi: 10.1109/SmartNets58706.2023.10216147.

[8] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, “Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems,” *J. Inf. Secur. Appl.*, vol. 58, 2021, doi: 10.1016/j.jisa.2020.102717.

[9] B. Biggio *et al.*, “Evasion attacks against machine learning at test time,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8190 LNAI, no. PART 3, pp. 387–402, 2013, doi: 10.1007/978-3-642-40994-3_25.

[10] H. Lin, J. Zhuang, Y.-C. Hu, and H. Zhou, “DefRec: Establishing Physical Function Virtualization to Disrupt Reconnaissance of Power Grids’ Cyber-Physical Infrastructures,” no. February, 2020, doi: 10.14722/ndss.2020.24365.

[11] J. H. Jafarian, E. Al-Shaer, and Q. Duan, “An Effective Address Mutation Approach for Disrupting Reconnaissance Attacks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 12, pp. 2562–2577, 2015, doi: 10.1109/TIFS.2015.2467358.

[12] P. Keshavamurthy and S. Kulkarni, “Early Detection of Reconnaissance Attacks on IoT Devices by Analyzing Performance and Traffic Characteristics,” *2023 IEEE Int. Conf. Cyber Secur. Resil.*, pp. 187–193, 2023, doi: 10.1109/csr57506.2023.10224986.

[13] M. Lyu, H. H. Gharakheili, and V. Sivaraman, “A Survey on Enterprise Network Security: Asset Behavioral Monitoring and Distributed Attack Detection,” pp. 1–18, 2023, [Online]. Available: http://arxiv.org/abs/2306.16675.

[14] V. Q. Nguyen, T. L. Ngo, L. M. Nguyen, V. H. Nguyen, V. Van Nguyen, and T. H. Nguyen, “Hybrid of Deep Auto-Encoder and Maximum Mean Discrepancy for Cyber Reconnaissance Detection,” *2023 15th Int. Conf. Knowl. Syst. Eng.*, pp. 1–8, 2023, doi: 10.1109/kse59128.2023.10299465.

[15] A. C. for C. S. (ACCS), “UNSW_NB15.” https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15?rvi=1.

[16] V. Kumar, “Feature Selection: A literature Review,” *Smart Comput. Rev.*, vol. 4, no. 3, 2014, doi: 10.6029/smartcr.2014.03.007.

[17] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, “SCADA system testbed for cybersecurity research using machine learning approach,” *Futur. Internet*, vol. 10, no. 8, 2018, doi: 10.3390/fi10080076.