**Journal of Information Systems Engineering and Management**
2025, 10(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

# Enhancing Cybersecurity in Wireless Sensor Networks: AI Solutions to Simulated Attacks

Mohit Angurala[1], Mandeep Kaur[2], Varinder Pabbi[3], Nidhi Chopra[4]

[1,2]*Computer Science, Guru Nanak Dev University College (Constituent College of Guru Nanak Dev University, Amritsar), Pathankot, Punjab, india*

[3]*Computer Science and Engineering, 2Computer Application, Ramgarhia Institute of Engg. & Tech. Phagwara. Punjab, India*

[4]*Department of Information Technology, Lyallpur Khalsa College Technical Campus, Jalandhar, Punjab, India*

*Corresponding Author Email: [4]chopranidhi44@gmail.com*

**ARTICLE INFO**

Received: 06 Oct 2024

Revised: 02 Dec 2024

Accepted: 18 Dec 2024

**ABSTRACT**

Owing to the real time applications of Wireless Sensor Networks (WSNs) including: industrial automation and remote environment monitoring, WSNs have revolutionized today's infrastructure. While implementing WSNs in strategic areas, security threats have become increasingly prevalent. Security enhancement in WSN by adopting advanced techniques in machine learning is the major focus of this research work. In an effort to discover possible use of Random Forest and Isolation Forest algorithms on them to detect and prevent the attacks, we look into depth of the attack. In this paper, the dataset is pulled from different repositories that are freely available to the public as an initial step followed by various preprocessing techniques. Data cleaning, feature selection, normalization, and categorical variable encoding have been applied as a part of preprocessing. We then observed a general increase in the detection of malicious flows together with the improvement of the tolerance to the simulated attacks. Moreover, we observed how ML enhances the security of WSNs with the combined use of ensemble learning and anomaly detections showing promising approaches and foundations for theoretical and experimental studies. The carried-out experiment proves the efficacy of the Random Forest Classifier (RFC), while maintaining a high level of accuracy, which is 99. 86% compared to 99. 72% before the attack.

**Keywords:** Wireless Sensor Networks (WSNs), Machine Learning, Random Forest Classifier (RFC), Isolation Forest, Network Security.

## INTRODUCTION

Wireless Sensor Networks (WSNs) have incredible flexibility and have indeed changed the core of today's infrastructures. In a network, a large number of sensor nodes can indicate alterations in their physical or environment context and acquire data and forwards the information to the base station. Despite the fact that WSNs are very efficient and contribute significantly to the information decision-making process in many global industries, there are significant security concerns when deploying an WSN in sensitive and critical environments.

It can be noted that WSNs are vulnerable to certain attacks. The complexity of these attacks can therefore be categorized into simple Interception and manipulation, to more complex attacks whose aims are to disrupt the network operations or corrupt data. Preserving infrastructures from such attacks in WSNs is paramount because these are usually placed in environments which are inherently distributed. In general, over the past few years, methods based on Machine Learning (ML) have become essential tools when it comes to enhancing the WSNs' security and other technological fields. For threat detection and response in real-time, ML has a better ability of anomaly detection, categorization, and predictive modeling. That is, with reference to the data patterns, ML is characterized by its ability to learn and adapt. Among these ML methods, Random Forests and other ensemble learning methods are the most useful for WSNs in distinguishing between the malicious activity and normal network traffic.

Based on the modern approaches in the field of ML, this paper investigates the possibilities of enhancing the security of WSNs. First of all, the data from various sources are collected and further cleaned and prepared very carefully with reference to some features influencing the operation and protection of WSNs. Next, to help in detecting different kinds of threats or attacks, Random Forest classifiers are applied to search for the hidden connections in the data set. To enhance the system's security from threats, we use Isolation Forest method, which is widely used for isolating and detecting any loopholes that might mean possible attacks.

## RELATED WORK

In the case of monitoring environmental conditions or industrial automation, WSN becomes inducive. Depending on their flaws and limited resources, they offer significant security threats nonetheless used extensively.

Securing WSN from threats or attacks has been a challenging task for many researchers. Kalita and Kar [1] described some of the challenges like communication attack and node attack, and they both called for heightened security measures. By modifying the modulation techniques of MAODV in WSNs, the performance of the Solar-MAODV model was analyzed by Angurala et al. [2], but lack of security is one of the major issues in the proposed model which leads to performance issues in WSNs. The concern was majorly focussed upon the usage of energy. Through enhancement of sustainability and operational durability, Angurala, Bala, and Bamber [3] were able to propose an efficient load balancing approach for energy restoration in WSNs. However, the security feature ignored by them again affected the performance of the proposed model for WSNs.

In an attempt to raise the reliability of detecting possible attacks in sensor networks, Rajasegarar et al. [4] offered Centered Hyperspherical and Hyperellipsoidal One-Class Support Vector Machines (SVMs). Hemalatha et al. [5] further compared the efficiency of SVMs' linear and quadratic programming and concluded that the SVM-based security solutions proposed for WSN are effective in the classification tasks, with high accuracy rates achieved through the real world datasets. Vinayakumar et al. [6] proposed intelligent intrusion detection in context of WSNs as a deep learning solution that compares with neural networks to enhance the detection accuracy and flexibility in relation to emerging threats. Patel and Mistry [7] conducted the evaluation of the detection strategies of the Sybil attack in WSNs.

Angurala and Bharti [8] compared LEACH and PEGASIS and concluded that LEACH can perform better than PEGASIS in many performance metrics including security and performance. In regards with security concerns on the major layer of WSNs, which is the perception layer, Zhang et al. [9] proposed a digital watermarking method useful in defending the integrity of data in IoT. For enhancing the security of WSNs, Yi et al. [10] introduced a block encryption algorithm recognized as a chaotic S-Box that performs encryption in order to defend the useful data. Network design should focus on the security aspect as suggested by Patel and Kumar [11]. Zhang, Heys, and Li [12] explained the trade-offs between energy consumption and security measures by conducting the energy assessment of encryption algorithms used on the WSN. Luo et al., [13] proposed a cooperative target tracking system that integrates WSN and WiFi (Wireless Fidelity) in order to address the problems of wireless network security. Specifically, they focussed on the issues that may occur with the security of industrial Internet of Things (IoT) settings. Qiao and Ma [14] improved the ZigBee-based WSNs, utilizing Bluetooth in industrial filed measurement by concentrating on the communication. Integrity and confidentiality of data in interconnected sensors was cited as one of the primary challenges in the IoT privacy and security by Yang et al. in their survey [15]. A study done by Yu et al. [16] concerning the assessment of the WSN security requirements pointed out that because of the specific nature of these networks, specialized solutions are needed for robust operations because of the specific security risks.

For improving methods of energy replenishment for sustainable WSNs to address operational problems in the long run, Akhtar and Rehmani [17] studied the integration of renewable and non-renewable energy. In historical and current perspective of state of the art and future trends of WSN security and privacy, Lee [18] worked on the solutions for new prevailing threats. In one of the work Wood and Stankovic [19] focused on denial of service (DoS) attacks and methods of mitigating the impact of these threats. To the highlights of trust, privacy, and security issues in the IoT systems, Sicari et al. [20] proposed an advancement of security levels to develop more reliability and confidentiality in the connected sensors' networks. To reduce the impact of DoS attacks on WSN performance, the work of Raymond and Midkiff [21] focuses on defending and attacking strategies in WSN.

Table 1: Existing Methods Comparison Chart with Important Results.

| Reference | Technique/Methodology | Focus Area | Year | Key Findings |
|---|---|---|---|---|
| [1] | Review of vulnerabilities and threats | Security Analysis | 2009 | Identified vulnerabilities in WSNs |
| [2] | Solar-MAODV, modulation techniques | Energy Efficiency | 2022 | Improved energy efficiency in WSNs |
| [3] | Novel techniques for load balancing | Energy Replenishment | 2021 | Enhanced sustainability in WSNs |
| [4] | One-class SVM | Anomaly Detection | 2010 | Effective anomaly detection |
| [5] | SVM variations | Machine Learning | 2009 | Optimization of SVM-based security solutions |
| [6] | Deep learning | Intrusion Detection | 2019 | Intelligent intrusion detection in WSNs |
| [7] | Sybil attacks | Attack Detection | 2017 | Techniques for detecting Sybil attacks |
| [8] | LEACH vs. PEGASIS | Protocol Comparison | 2016 | Comparative study on energy efficiency |
| [9] | Digital watermarking | Data Integrity | 2017 | Protection of data integrity in IoT |
| [10] | Chaotic S-Box | Encryption | 2019 | Novel encryption algorithm for WSNs |
| [11] | Challenges and prospects | Future Prospects | 2018 | Future directions for WSN security |
| [12] | Encryption schemes | Energy Efficiency | 2012 | Energy-efficient security measures |
| [13] | WSN and WiFi integration | Target Tracking | 2018 | Reliable target tracking in indoor networks |
| [14] | ZigBee with Bluetooth | Communication Protocol | 2015 | Enhanced communication protocols |
| [15] | IoT survey | Security Challenges | 2017 | Challenges in IoT security |
| [16] | WSNs characteristics | Security Requirements | 2020 | Tailored security requirements for WSNs |
| [17] | Renewable energy integration | Energy Replenishment | 2015 | Sustainable energy solutions for WSNs |
| [18] | Recent developments | Advances in Security | 2020 | Advances and challenges in WSN security |
| [19] | Impact and defenses | Denial-of-Service | 2002 | Strategies against DoS attacks in WSNs |
| [20] | Privacy and trust issues | IoT Security | 2015 | Ensuring privacy and trust in IoT networks |
| [21] | Countermeasures | DoS Attacks | 2008 | Defense mechanisms against DoS attacks |

WSNs are not easy to protect and enhancing WSNs security entail addressing some of the challenges that include; solving vulnerabilities, energy management, communication, and data (Refer Table 1). One possible approach is to enhance the security of WSN employing the Random Forest algorithms that is depends on the ensemble learning techniques. Thus, Random Forest is perfectly suitable for solving such tasks which includes: anomaly detection, prevention of intrusions, or classification since it can operate on vast volumes of data, and recognize essential attributes. Immune to new threats or attacks, and faster speed of the network are the outcomes resulting from

applying Random Forest on WSN security frameworks. These are the challenges that affects WSN operations' security and reliability; therefore, its adaptive learning capabilities and ability to handle noisy data are critical [1].

## MATERIALS AND METHODS

In this section, demonstration has been conducted to prove that our method is more accurate in detection while, at the same time, it is more robust under simulation attack. The work formulates, how ML provides improved protection to WSNs against new forms of security threats through the integration of ensemble learning along with current ways of anomaly detection. The details of a vast look at our study's methodological design are presented in the next section. The design includes: data extraction from datasets, cleaning, training and testing procedures, selection of an appropriate model and the specification of a study, quantitative results of the study. We also discuss our advancements in the said field, explain the implications of our results, and identify more research directions to enhance the stability of WSN security frameworks. Following is the list of procedures outlined for the carried-out proposed method:

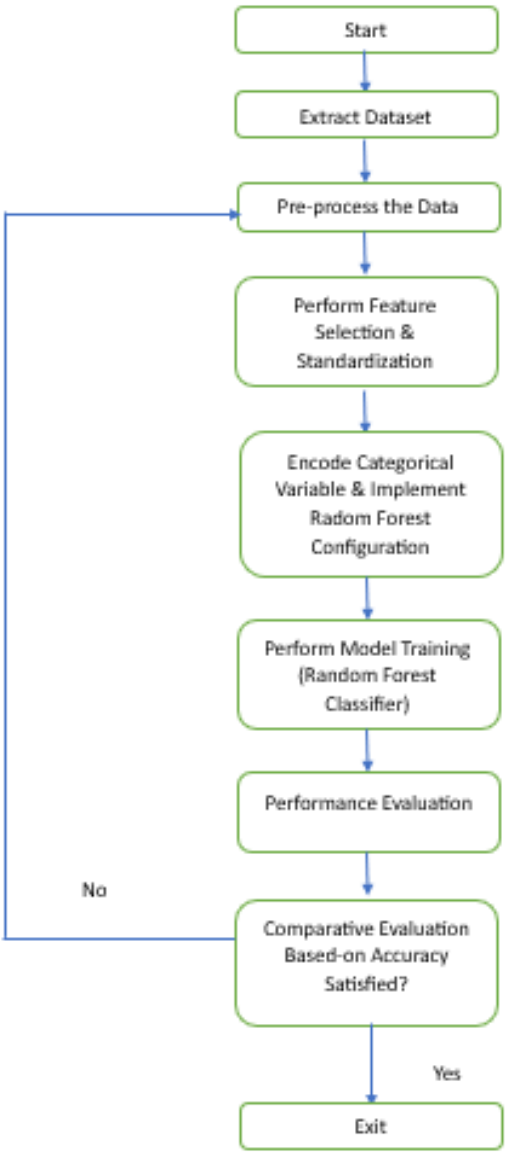


Figure 1: Methodology for the Intrusion Prevention System for WSNs.

Essential aspects of our proposed design include the first step of data gathering from various sources, particularly for databases related to WSNs. The specific aspects of WSN and the relative security parameters like the sensed data, and statistical data of the previous and past activities are considered. This dataset is a necessity for the artificial intelligence (AI) to differentiate between normal network traffic and the other forms of attacks.

While extracting the data, a systematic approach has been applied so that none of the data gets lost or duplicated in the process. The dataset is also required to be filtered on the basis of its accuracy, completeness, and its relevance to our set research objectives. In order to ensure the stability and quality of the dataset for the rest of the research, all the inconsistencies or sometimes lack of data is well noted and rectified through data validation techniques.

**Preprocessing Steps:**

After getting the dataset, the subjected data passes through certain rigorous preprocessing to make the data well prepared for analysis as well as model training. This phase is significant since enhancing the data quality, reducing errors, and improving the performance of the ML algorithms are significant. The primary procedures for preprocessing are:

1. **Data Cleaning:** The first action is to clean the dataset to ensure no wrong and/or incomplete information exist in the dataset. In order to keep the dataset complete and reasonably accurate, some data filling techniques, for example, imputation, are applied in order to fill in missing values with estimated average values such as average, median, or mode.
2. **Focusing on the Most Relevant Variables for Research:** Due to the large amount of data, which is assigned to model as input, do feature selection. They are used in feature selection which maintains the most important variables out of the enormous data set. It enhanced the effectiveness of the computational processes and provided more focus on important aspects of the models inside the communication network to differentiate between the standard behaviour from malicious attacks.
3. **Normalization and Scaling:** Normalization and Scaling are important because no specific feature's numerical values take over during model training.
4. **Encoding Categorical Variables**: The last data preparation step is also the most straightforward due to the categorial nature of some variables such as attack types and sensor IDs – these values had to undergo one-hot encoding as are essential to be read by ML algorithms. In order to incorporate these variables into the training of the model, one had to use the procedures like one-hot encoding or label encoding depending on the data and its characteristics.

It can be noted that all the pretreatment activities are carried out very diligently, ensuring that the dataset is prepared well for evaluation and application of our models in the subsequent steps. As we are going to observe in the later sections of this paper, these preprocessing measures provided a good ground work in employing complex ML algorithms. The approaches that have been used in this research are mainly based on Random Forest classifier to distinguish between normal network traffic and potential threats in WSNs. This assembling learning design is very systematic, which brings out good performance without the danger of over training and it is highly acclaimed as being very effective in solving complex classification problems. During the implementation phase, the following configuration specifications were used:

**Table 2:** Random Forest Classifier Configuration

| Parameter | Value | Description |
|---|---|---|
| n_estimators | 100 | The number of decision trees comprising the forest influences both model robustness and computational efficiency. |
| Criterion | 'gini' | The measure of split quality at each node can be assessed using options such as Gini impurity ('gini') or information gain ('entropy'). |
| max_depth | None | A simple solution to addressing the mechanism's complexity and reducing the risk of overfitting in a Random Forest classifier is to limit the depth of decision trees. One more constraint that regulates the possible depth of the decision tree during training of samples is the maximum number of levels in it. |

| | | |
|---|---|---|
| min_samples_split | 2 | In a Random Forest classifier, nodes will not be split any further if the sample size is smaller than minimum needed to split an internal node in a decision tree. |
| min_samples_leaf | 1 | A number of samples in a node of a decision tree in the Random Forest classifier is always not less than the number of samples in the terminal node. |
| max_features | auto | A square root of the total number of features is often artificially set to equal to 1 while searching in a Random Forest classifier for the best split at a certain node in a decision tree. |

**Model Training Process**

In order to train the Random Forest classifier to accurately distinguish between normal and unusual network activity, a number of rigorous processes are involved:

**Data Partitioning:** The independent dataset is split into 80/20 where in the division, X_train, y_train has been used for training and X_test, and y_test for the testing of the dataset. After that, regarding the fairness of model assessment, our approach considered relative proportions of classes (normal vs. attacks) in both datasets.

**Model Fitting:** To accommodate the model, we first introduced the training data as X_train, y_train and after that we initiated as well as trained Random Forest classifier on both of these data matrices. During this stage, the classifier fitted decision trees on features (X_train) and the labels which refers to the targets (y_train). To correctly distinguish between "Normal" label (0) and attack instances (1) it aimed to enhance split criteria.

**Performance Evaluation:** According to the testing data (X_test, y_test), we maintained the same perspective of 80/20 splitting ratio and class distribution. Hence, based on the ability to solve complex problems of categorization the Random Forest was chosen.

**Algorithm:**

The proposed security mechanism integrates Random Forest and Isolation Forest algorithms to enhance the resilience of Wireless Sensor Networks against security threats.

**Algorithm 1: Secure WSNs with the Help of Random Forest and Isolation Forest Algorithm**

**Input: Dataset D containing features X and labels y**

**Output: Results: Wireless sensor networks are better able to identify and avoid threats.**

**Step 1: Dataset Extraction and Preprocessing**

**1.1. Find a public repository and get dataset D from there.**

**1.2. Clean the data:**

**Identify and handle missing or incorrect data points.**

**Use methods like imputation to fill missing values.**

**1.3. Select features:**

**Perform correlation analysis to retain significant features.**

**1.4. Normalize and scale the dataset:**

**Apply min-max normalization to standardize feature scales.**

**1.5. Encode categorical variables:**

**Utilize one-hot encoding for categorical features.**

**Step 2: Model Implementation**

**2.1. Split dataset D into training set (X_train, y_train) and testing set (X_test, y_test) using an 80-20 stratified split.**

**2.2. Initialize and configure the Random Forest classifier RF with parameters listed in Table 1.**

**2.3. Train RF on the training set (X_train, y_train).**

**Step 3: Attack Simulation and Anomaly Detection**

**3.1. Introduce Gaussian noise to simulate attacks in the testing set X_test.**

**3.2. Initialize and apply the Isolation Forest IF to detect anomalies in the noisy test data.**

**3.3. Filter out anomalies detected by IF to obtain the filtered test set (X_test_filtered, y_test_filtered).**

**Step 4: Model Evaluation**

**4.1. Evaluate RF on the original testing set (X_test, y_test) to compute initial accuracy, precision, recall, and F1 Score.**

**4.2. Evaluate RF on the filtered test set (X_test_filtered, y_test_filtered) to compute metrics after filtering out anomalies.**

## SIMULATION RESULTS AND PERFORMANCE ANALYSIS

Configured with 100 decision trees and using the Gini impurity criterion for node splitting, the model was trained on the training data to understand patterns and relationships among features. When evaluating the model's performance on the test set, we found that it achieved an initial accuracy of about 99.72% (Refer Fig. 2). This accuracy metric reflects how well the model can correctly classify instances of both 'Normal' behavior and different types of attacks based on what it learned during training. Subsequently, we noisy the test data with Gaussian noise in order to simulate an attack scenario and examine the model's resistance to possible dangers. Here, we suggest realistic examples in which the sensors fail or the data integrity is in danger in actual installations. After the introduction of this noise, an Isolation Forest method is applied to filter out outliers from the mentioned test set. To make the model effective in not taking wrong decisions when the data is disturbed or corrupted, there is an additional phase added in anomaly detection. The accuracy of filtered numbers was is raised to about 99. 86% after these attack mitigation steps show a little more enhancement than the initial accuracy. This improvement shows that current classifiers perform well when incorporated with detection strategies such as the Isolation Forest. It enhances the model's antifragility that enables it to defend itself in simulations and the capability to sustain functionality under suboptimal circumstances.

### Accuracy

The accuracy of a model reveals the frequency with which it correctly predicts all possible outcomes.

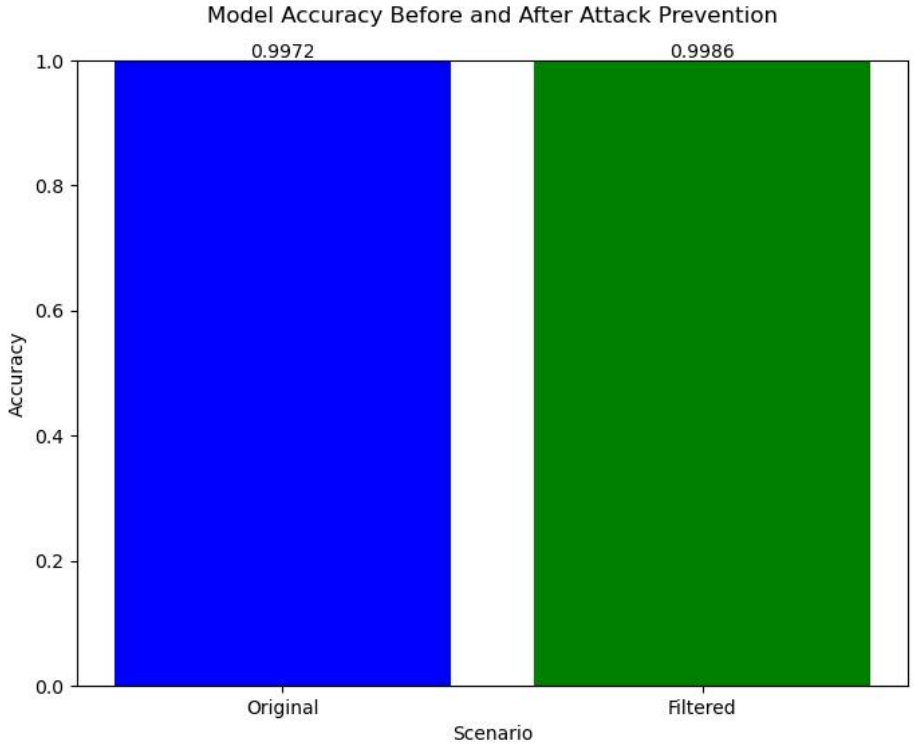


Figure 2: Accuracy Comparison of Model.

**Before Attack Prevention:**

This model accurately defined most of the cases in the dataset for normal and attack conditions with the primary accuracy of up to 99. 72% (Refer eq. 1). Such levels of differentiation can render the fundamental concept of the model robust, given it can distinguish normal network traffic from threat activities.

$$Accuracy=(TP+TN)/(TP+TN+FP+FN) \quad (1)$$

**After Attack Prevention:**

From equation (2) it can be seen that with the help of attack avoidance measures and using Isolation Forest, the filtered accuracy rose to 99. 86%. This enhancement is due to the fact that, when selecting test data, it is only natural to exclude any outliers that may distort the results. In this way, the model reached the higher accuracy of the outcomes; thereby leaving such cases which are considered as the outliers have contributed the model but did not allow it to focus on the cases identification.

$$\text{Filtered Accuracy}=(〖TP〗\_\text{Filtered}+〖TN〗\_\text{Filtered})/(〖TP〗\_\text{Filtered}+〖TN〗\_\text{Filtered}+〖FP〗\_\text{Filtered}+〖FN〗\_\text{Filtered}) \quad (2)$$

**Precision**

The precision is calculated with the help of the ratio of the number of attacks identified by the model to the number of correct outcomes.

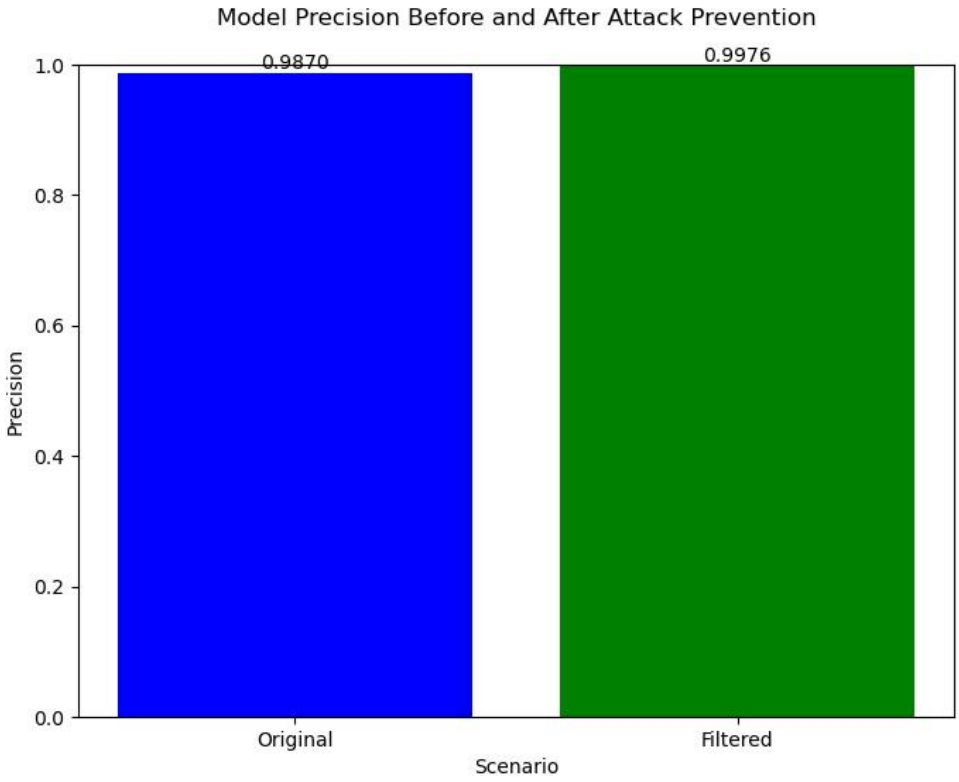


Figure 3: Precision Comparison of Model.

**Before Attack Prevention:**

A precision of 98.70 percent is noticed when no attack is launched. However, it means that it had correctly identified attacks approximately 98 percent of the time as stated in equation (3). Since few normal connections are considered as an attack, the accuracy rate is high in this case.

$$\text{Precision}=TP/(TP+FP) \quad (3)$$

**After Attack Prevention:**

Refer to equation (4), it is observed that there is a substantial increase in the accuracy to 99%. This enhancement shows that indeed the Isolation Forest was able to correctly isolate and remove what was most likely to be an abnormality from the data set. Consequently, false positive attack predictions became less frequent and the model's attack prediction capability increased.

$$\text{Filtered Precision}=〖TP〗\_\text{Filtered}/(〖TP〗\_\text{Filtered}+〖FP〗\_\text{Filtered}) \quad (4)$$

**Recall**

The ability of the model to identify the percentage of real-world positive events (attacks) is referred to as recall, or sensitivity.

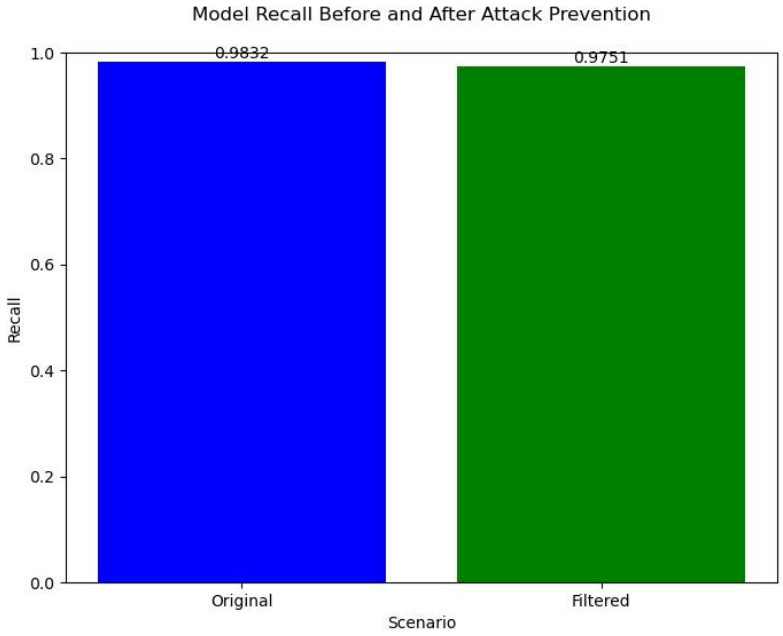


Figure 4: Recall Comparison of Model.

Before Attack Prevention:

Using eq. Given in (5), the model was able to successfully identify with 98% accuracy, which means model's recall is 98%. A model therefore achieved high recall owing to few false negatives.

$$\text{Recall}=TP/(TP+FN) \quad (5)$$

**After Attack Prevention:**

Using eq. (6) the recall decreased up to 97.51%. While Isolation Forest is able to erase noise, this decrease in effectiveness means that the model is somewhat less effective at identifying all attacks. However, the trade-off shows that although noise reduction improves overall data, there is a problem with the model's ability to detect any potential attacks

$$\text{Filtered Recall}=〖TP〗\_\text{Filtered}/(〖TP〗\_\text{Filtered}+〖FN〗\_\text{Filtered}) \quad (6)$$

**F1 Score**

The F1 Score is also useful because it considers not only true positives but also false negatives while evaluating the model's performance.

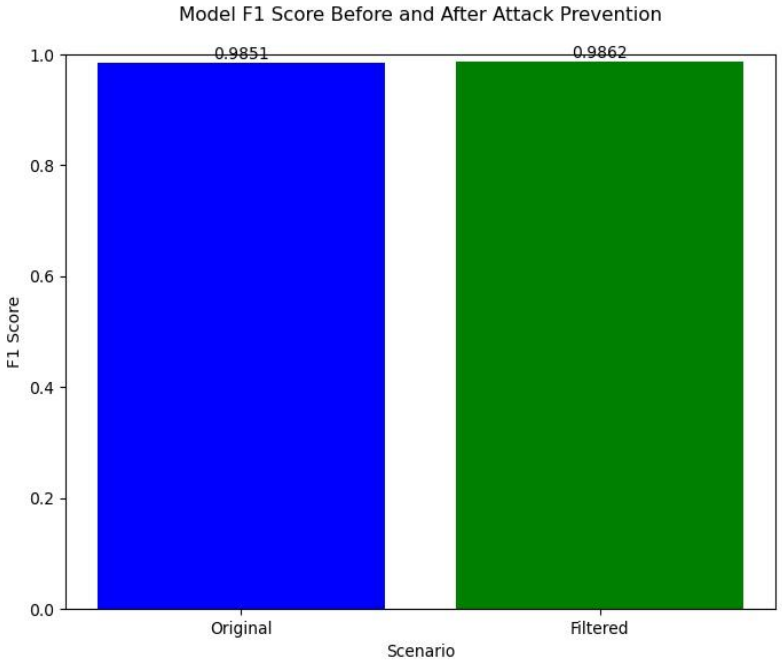


Figure 5: F1 Score Comparison of Model.

**Before Attack Prevention:**

The model was able to show equal and considerable performance by starting with an F1 Score of 98.51%, based on equation (7). They were able to demonstrate good memory as well as the level of precision of their work. Indeed, this shows that the model is capable of differentiating between normal or acceptable events and those that are unwanted or malicious.

$$\text{F1 Score} = 2^* \ (\text{Precision}^*\text{Recall})/(\text{Precision}+\text{Recall}) \quad (7)$$

**After Attack Prevention:**

As to the overall improvement of the results from the attack prevention measures, the F1 Score rose to a percent of 98.62% (refer eq. (8)). The above enhancement clearly proves that though there was a slight decline in recall, the increase in the accuracy made up for it. The model's overall performance is enhanced when it is about correctly identifying occurrence of normal and attack after Isolation Forest preprocessing.

$$\text{Filtered F1 Score} = 2^*(\text{Filtered Precision} * \text{Filtered Recall})/(\text{Filtered Precision} + \text{Filtered Recall}) \quad (8)$$

The improvements or declines in the accuracy, precision, recall and F1 Score in cybersecurity applications before and after the measures such as avoiding the attacks are the signs to explain the effect of data preparation on the modelling. The use of Isolation Forest yielded satisfactory results which indicate that the degree of accuracy in identifying a set of instances was improved by the function of disregarding possible outliers or anomalies from the given set. Even though there is a slight decrease in the recall when compared to the previous experiments, F1 Score and the accuracy of the model showed a large increment of the overall total performance. These additions demonstrate that the attack prevention strategies are effective in fortifying the model's robustness and dependability. They demonstrate the need to always pre-process data before feeding it to machine learning models especially on sensitive areas such as cybersecurity anomaly detection.

## CONCLUSION

Descriptive statistics of F1 Score, recall, accuracy, and precision changes before and after the implementation of attack prevention measures demonstrate how data pretreatment affects the performance on cybersecurity applications. The latter provided the model with the possibility of improving the accuracy of correct identification of occurrences after excluding typical outliers or anomalies from the dataset, as can be seen after applying Isolation Forest. With an understanding that recall might be an issue with the larger dataset, overall performance

measurements captured accuracies' enhancements as well as the F1 Score. This demonstrates how attack avoidance strategies help ensure the model is developed to be less vulnerable. To important aspects in tender areas, such as cybersecurity anomaly detection, they accentuate the necessity of adequate data pretreatment for ensuring ML's model capacity and efficiency.

## REFERENCES

[1] H. K. Kalita and A. Kar, "Wireless sensor network security analysis," International Journal of Next-Generation Networks, vol. 1, no. 1, pp. 01-09, Dec. 2009.

[2] M. Angurala, H. Singh, Anupriya, A. Grover, and M. Singh, "Testing Solar-MAODV energy efficient model on various modulation techniques in wireless sensor and optical networks," Wireless Networks, vol. 28, pp. 413-425, 2022.

[3] M. Angurala, M. Bala, and S. S. Bamber, "A novel technique for energy replenishment and load balancing in wireless sensor networks," Optik, vol. 248, p. 168136, 2021.

[4] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, "Centered Hyperspherical and Hyperellipsoidal One-Class Support Vector Machines for Anomaly Detection in Sensor Networks," IEEE Transactions on Information Forensics and Security, vol. 5, no. 3, pp. 518-533, Sep. 2010.

[5] Hemalatha. T, S. Mithila. T, and Soman. K. P, "Comparative study of linear and quadratic programming versions of SVM on various real life data sets," International Journal of Recent Trends in Engineering, vol. 1, no. 2, pp. 23-25, May 2009.

[6] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., and Venkatraman, S., "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[7] Patel, S. T. and Mistry, N. H., "A review: Sybil attack detection techniques in WSN," in Proceedings of the 2017 4th International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, pp. 184–188, Feb. 2017.

[8] M. Angurala and Bharti, "A comparative study between LEACH and PEGASIS—A review," in IEEE Conference on Computing for Sustainable Global Development, 2016, pp. 3271-3274.

[9] Zhang, G., Kou, L., Zhang, L., Liu, C., Da, Q., and Sun, J., "A New Digital Watermarking Method for Data Integrity Protection in the Perception Layer of IoT," Security and Communication Networks, vol. 2017, pp. 3126010, 2017.

[10] Yi, L., Tong, X., Wang, Z., Zhang, M., Zhu, H., and Liu, J., "A novel block encryption algorithm based on chaotic S-Box for wireless sensor network," IEEE Access, vol. 7, pp. 53079–53090, 2019.

[11] Patel, N. R. and Kumar, S., "Wireless Sensor Networks' Challenges and Future Prospects," in Proceedings of the 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, pp. 60–65, Nov. 2018.

[12] Zhang, X., Heys, H. M., and Li, C., "Energy efficiency of encryption schemes applied to wireless sensor networks," Security and Communication Networks, vol. 5, pp. 789–808, 2012.

[13] Luo, J., Zhang, Z., Liu, C., and Luo, H., "Reliable and Cooperative Target Tracking Based on WSN and WiFi in Indoor Wireless Networks," IEEE Access, vol. 6, pp. 24846–24855, 2018.

[14] Qiao, B. and Ma, K., "An enhancement of the ZigBee wireless sensor network using bluetooth for industrial field measurement," in Proceedings of the 2015 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP), Suzhou, China, pp. 2–4, Jul. 2015.

[15] Yang, Y., Wu, L., Yin, G., Li, L., and Zhao, H., "A Survey on Security and Privacy Issues in Internet-of-Things," IEEE Internet of Things Journal, vol. 4, pp. 1250–1258, 2017.

[16] Yu, J. Y., Lee, E., Oh, S. R., Seo, Y. D., and Kim, Y. G., "A Survey on Security Requirements for WSNs: Focusing on the Characteristics Related to Security," IEEE Access, vol. 8, pp. 45304–45324, 2020.

[17] Akhtar, F. and Rehmani, M. H., "Energy replenishment using renewable and traditional energy resources for sustainable wireless sensor networks: A review," Renewable and Sustainable Energy Reviews, vol. 45, pp. 769–784, 2015.

[18] Lee, C. C., "Security and privacy in wireless sensor networks: Advances and challenges," Sensors, vol. 20, no. 744, 2020.

[19] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," IEEE Journal, vol. 35, no. 10, pp. 54-62, Oct. 2002.

[20] Sicari, S., Rizzardi, A., Grieco, L. A., and Coen-Porisini, A., "Security, privacy and trust in Internet of Things: The road ahead," Computer Networks, vol. 76, pp. 146–164, 2015.
[21] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," IEEE Pervasive Computing, vol. 7, no. 1, pp. 74-81, Mar. 2008.