International Research Journal of Engineering Science, Technology and Innovation Vol. 11(1) pp. 1-13, January, 2025
Available online http://www.interesjournals.org/IRJESTI

Research Article

Comprehensive Analysis of Wireless Sensor Network Attack Prediction Using Supervised Machine Learning Technique

Joy Winnie Wise DC, Revanth SB*, Sanjay D

Department of Chemical Engineering, Computer Science Engineering Rajalakshmi Institute of Technology, Chennai, India

*Corresponding Author's E-mail: revanth.s.b.2020.cse@ritchennai.edu.in

Received: 08-May-2024; Manuscript No: irjesti-24-134339; **Editor assigned:** 10-May-2024; Pre-QC No: irjesti-24-134339 (PQ); **Reviewed:** 24-May-2024; QC No: irjesti-24-134339; **Revised:** 08-Jan-2025; Manuscript No: irjesti-24-134339 (R); **Published:** 15-Jan-2025, DOI: 10.14303/2315-5663.2025.99

Abstract

The wireless sensor network represents a paradigm of infrastructure-less networking, wherein the absence of centralized access points, routers, or servers is a defining characteristic. Rather than relying on such traditional infrastructure, this network model leverages nodes as the primary agents for transmitting data packets. The contemporary landscape of wireless sensor networks is marked by a myriad of challenges, foremost among them being data security. As the proliferation of these networks continues, they are increasingly susceptible to a diverse array of attacks aimed at compromising data transmission integrity and precipitating data loss. Denial-of-Service (DoS) attacks pose a significant threat by inundating the network with spurious requests or traffic, thereby impeding legitimate communication and disrupting network functionality. Additionally, node compromise attacks exploit vulnerabilities within individual nodes, enabling adversaries to gain unauthorized access and exert control over critical network components. The ability to predict and prevent these attacks is crucial for maintaining a secure network environment. Our study offers a thorough examination of supervised machine learning methods for predicting network attacks. We gather and preprocess data, extracting pertinent features and formatting them for machine learning algorithms. We assess the effectiveness of these algorithms and explore the interpretability of the trained models to uncover insights into the patterns and traits of network attacks. This enables network administrators to grasp the attack landscape and devise tailored defense strategies.

Keywords: Machine learning, Wireless sensor networks, Cybersecurity, Machine learning algorithms, Network environment

INTRODUCTION

Wireless Sensor Networks (WSNs) have become very adaptable instruments in several domains, providing an array of benefits that transform automation, control, and monitoring procedures (Salmi S et al., 2023). Wireless Sensor Networks (WSNs) provide real-time data collecting, analysis, and decision-making in a variety of applications, from industrial automation to environmental monitoring (Wazirali R et al., 2022). A wireless sensor network operates on the principle of data transmission devoid of physical tethering, employing wireless protocols like WiFi and Bluetooth for information exchange. Within this decentralized infrastructure, numerous nodes interconnect sans central administrative oversight, allowing any device to assume the role of a node (Ismail S et al., 2022). The absence of stringent verification or security measures exposes the network to the potential infiltration of malicious nodes, precipitating risks such as denial of service, data packet interception, and data compromise (Ifzarne S et al., 2021).

Numerous studies have delved into wireless sensor network attacks, leading to the development of various machine learning models aimed at predicting these attacks (Alsulaiman L et al., 2021). However, a significant challenge lies in the availability of data, particularly since wireless sensor networks often contain highly sensitive information inaccessible to the public (Tabbaa H et al., 2022). This scarcity of datasets makes it difficult to train robust machine learning models capable of accurately predicting these attacks (Gebremariam GG et al., 2023).

To address this challenge, our study introduces multiple machine learning models trained on a dataset comprising over 300,000 records sourced from Kaggle, an online community of data scientists and machine learning engineers (Alsahli MS et al., 2021). We emphasize the

interpretability of these machine learning models and conduct an in-depth analysis to identify models with high accuracy. Subsequently, we deploy selected model as a real-time application, enabling users to detect occurrences in wireless sensor networks by providing input based on the features outlined in the dataset (Baraneetharan E, 2020).

Related works

This section offers a thorough explanation of the research done before on preventing attacks in wireless sensor networks. It goes into detail about the pros and cons of those previous efforts and compares them to what we're doing now (Almomani I et al., 2016). This section additionally furnishes a comprehensive exposition on the imperative need for security in Wireless Sensor Networks (WSN), delving into the intricacies of machine learning techniques (Ashraf S et al., 2020). It further distinguishes between various methodologies employed in machine learning for preventing attacks in WSN, elucidating the datasets utilized in these endeavors (Feng X et al., 2013).

Overview

The security of Wireless Sensor Networks (WSN) constitutes a significant apprehension for the transmission of data packets (Kumar BS et al., 2022). Given WSN's widespread adoption as a cost-effective communication technology and its prevalence in numerous Internet of Things (IoT) devices, the decentralized nature of WSN poses particular challenges (Rao GS et al., 2023). Without adequate security measures, any wireless communication-enabled device can function as a node within the network, rendering it susceptible to intrusion by malicious entities (Tomic I et al., 2017). This vulnerability exposes WSN to various types of attacks, notably blackhole and grayhole attacks, thereby amplifying the need for robust security protocols and defenses (Shi E et al., 2004).

Machine learning

Wireless Sensor Networks (WSN) are susceptible to numerous types of attacks, underscoring the imperative for comprehensive understanding and prevention measures (Premkumar M et al., 2023). This section will delve into a detailed examination of the diverse array of attacks targeting WSN, elucidating their intricacies and potential ramifications (Pan JS et al., 2021). Additionally, it will underscore the significance of pre-emptive measures for preventing WSN attacks, emphasizing the critical importance of safeguarding these networks against potential threats to ensure their integrity and functionality (Dener M et al., 2022). Machine learning is poised to play a pivotal role in enhancing the security of Wireless Sensor Networks (WSN), as evidenced by the predominant focus of previous research on machine learning methodologies within this domain (Roman R et al., 2009). Across various phases of investigation, researchers have employed diverse machine learning algorithms and datasets to address WSN security challenges comprehensively. Notable datasets utilized in prior studies include NSL-KDD and WSN, which have served as foundational resources for training and experimentation. Furthermore, a range of machine learning models has been applied, including Deep Belief Networks (DBN), Support Vector Machines (SVM), and Artificial Neural Networks (ANN). Despite their efficacy, these models confront a significant obstacle in the form of limited datasets, posing a challenge to their optimal performance and effectiveness in addressing WSN security concerns.

In order to detect DoS attacks in WSNs, Salmi and Oughdir the authors created and deployed a few deep learning models. Convolutional Neural Networks (CNN), Dense Neural Networks (DNN), recurrent neural networks (RNN), and a model that combines the RNN and CNN architectures are some of these models. Using the WSN-DS dataset, all models were trained. Based on their findings, CNN fared better than the other models, with an accuracy rate of 98.79%. One notable limitation of this research lies in the increased overhead incurred by deep learning models in comparison to traditional machine learning approaches.

Using WSN-DS datasets for training and testing, Wazirali et al., the author assessed a number of classification models, including SVM, GBoost, KNN, DT, LR, MLP, LSTM, and NB. Of these, GBoost performed the best, with an accuracy rate of 99.6% when averaged performance metrics were taken over all WSN-DS datasets. Moreover, ensemble approaches incur more overhead. However, they are inadequate for the network with limited computational power.

To lessen cyberattacks in WSNs,Ismail et al., the author suggested a multilayer machine learning detection methodology. They formed the first layer with an NB algorithm and the second layer with a Light method. The suggested method demonstrated a 99.3% detection accuracy rate using WSN-DS datasets. The majority of machine learning models have surpassed the efficacy of the proposed approach, including our own proposal.

Ifzarne et al., the authors proposed a detection model based on incremental machine learning. This model was constructed using the information gain ratio feature selection approach and the online passive aggressive classifier. With the help of the WSN-DS dataset, it was trained to identify DoS attacks. The accuracy rate in the simulation findings was 96%. The possibility of various machine-learning models for identifying denial-of-service assaults was investigated by Alsulaiman et al., the authors. The models, which included RF, j48, NB, SVM, and NN, were trained and tested using the WSN-DS dataset. The findings show that RF performed better than the others, obtaining a 99.72% accuracy rate. Random Forest indeed imposes a higher computational overhead, making it challenging to implement within low-cost communication technologies. The complexity of random forest algorithms demands significant computational resources, which may strain the limited processing capabilities and energy constraints inherent in low-cost communication technologies. As a result, deploying random forest models in such environments can lead to

inefficiencies and hinder the practical feasibility of these solutions. Consequently, alternative machine learning techniques that are more light weight and resource-efficient may be more suitable for integration into low-cost communication technologies.

H Tabbaa et al., the author proposed two ensemble approaches for enhancing detection accuracy: the homogeneous ensemble approach and the heterogeneous ensemble approach. In the homogeneous ensemble approach, they utilized the Hoeffding Adaptive Tree (HAT) technique, which comprises ten models. Conversely, the heterogeneous ensemble approach employed an Adaptive Random Forest (RF) model alongside the Hoeffding Adaptive Tree (HAT) algorithm. To evaluate the effectiveness of these approaches, they conducted experiments using the WSN-DS dataset. The results revealed detection accuracy rates of 97.2% for the homogeneous ensemble approach and 96.84% for the heterogeneous ensemble approach. Both ensemble approaches mentioned, although requiring substantial computational power, yielded lower accuracy rates compared to our proposed approach.

A technique to detect multiple WSN attacks was presented by the author Gebremariam et al. WNS-DS is one of four datasets used for testing and training the proposed approach. The technique they use to detect DoS attacks gradually achieves an accuracy rate of 99.65% when using the WNS-DS dataset. In this study, the author employed four datasets to train and test the proposed scheme, designed to detect various classes of attacks, including Denial of Service (DoS) attacks. The datasets utilized for training the machine learning model encompass WSN-DS, UNSW-NB, CICID2018, and NSL-KDD. The chosen machine learning model is the Artificial Neural Network (ANN). However, a drawback of the proposed model is the computational expense associated with neural network algorithms compared to conventional machine learning approaches.

Mohammed S. Alsahli., the author of this study addresses security concerns within Wireless Sensor Networks (WSN) by proposing automated solutions to identify associated risks. The effectiveness of various machine learning algorithms is evaluated using two types of datasets: KDD99 and WSN datasets. The goal is to analyze and safeguard WSN networks by integrating Firewalls, Deep Packet Inspection (DPI), and Intrusion Prevention Systems (IPS), each tailored for comprehensive protection. Multiple testing methodologies, including cross-validation and percentage split, were explored to assess algorithm performance. Based on the findings, the most accurate algorithm with the least processing time was recommended for both datasets.

In this proposal, the author Dr. E. Baraneetharan endeavors to provide a comprehensive elucidation of the pivotal role played by machine learning algorithms within Wireless Sensor Networks (WSN). The network's lifespan can be extended by using machine learning to provide real-time solutions that optimize resource use. The procedure is made easier, more dependable, efficient, and affordable by its ability to process autonomously without the need for external programming. Machine learning algorithms have the speed and accuracy to handle complex data. The environment of a wireless sensor network is improved by the application of machine learning. Decentralized and distributed in nature, Wireless Sensor Networks (WSN) are a hybrid of several networks. The self-organizing and self-healing characteristics of sensor and sink nodes make up a Wireless Sensor Network (WSN). In addition, WSN is employed in additional military applications, monitoring of climate change, biodiversity and ecosystem preservation, and surveillance. Now-a-days, a huge development is seen in WSNs due to the advancement of electronics and wireless communication technologies, several drawbacks like low computational capacity, small memory, and physical vulnerability in the infrastructure of resources is necessary to demand source protections, where privacy is crucial. Sensor networks require machine learning approaches to minimize needless redesign in order to adapt to dynamic environments, which WSN monitors. A overview of machine learning approaches for WSNs offers a multitude of applications where security is of utmost importance. In order to prevent hackers or other intruders from stealing data, the WSNs system should have the ability to erase instructions.

MATERIALS AND METHODS

This section shall present an overview of our methodology. Our study aims to address the vulnerabilities in Wireless Sensor Networks (WSNs) when connected to the internet, employing a supervised machine learning approach. Machine learning represents a prevalent methodology for tackling contemporary challenges. The utilization of machine learning within the context of WSN issues is anticipated to significantly enhance WSN security. In this approach, we mitigate WSN attacks by detecting various types of attacks occurring during data packet transmission between nodes within this decentralized network. Detection is achieved through the training of datasets using our machine learning models. We ascertain the interpretability of these machine learning models to identify the model that exhibits the highest accuracy in detecting WSN attacks. Subsequently, we deploy the machine learning model with the highest accuracy into a real-time web application. This deployment allows users to detect potential attacks during data packet transmission between nodes by inputting relevant feature values into the application.

In our WSN dataset, we encompass four distinct types of WSN attacks: Black hole attack, gray hole attack, flooding, and TDMA. These attacks serve as the foundation for training our machine learning model. The differentiation between our dataset and those utilized in previous research endeavors lies primarily in the volume of records. Past research datasets often suffer from a lack of

records due to the confidential nature of the information contained therein. These data, being highly sensitive, are not readily accessible to the public. Our dataset, however, addresses these limitations by providing over 300,000 records across five classes and encompassing 16 features. These features serve as the basis for training three distinct machine learning models, thus enhancing the security of WSNs through a comprehensive and robust approach.

Our study encompasses the utilization of three distinct classifiers: The ridge classifier, random forest classifier, and BernoulliNB. These classifiers are trained using our WSN dataset, which comprises 18 features and 5 classes aimed at detecting the four types of attacks on WSNs. Each classifier is tasked with discerning patterns within the dataset to accurately identify potential attacks on WSNs. Through rigorous training and evaluation, these classifiers yield varying levels of accuracy in their detection capabilities. The efficacy of each classifier is determined by comparing the accuracy levels attained across the three models. This comparison aids in identifying the most efficient classifier for detecting WSN attacks on the internet, thereby contributing to the enhancement of WSN security protocols.

All experiments conducted in this study will be executed using Jupyter notebook, an application accessible through Anaconda Navigator. Anaconda Navigator offers a comprehensive suite of Conda packages, encompassing a wide array of tools utilized for data preprocessing, visualization, analysis, and model training. Within Jupyter notebook, Python libraries such as Pandas and NumPy will be employed for data preprocessing tasks. Matplotlib and Seaborn will facilitate data visualization, enabling the clear representation of insights derived from the datasets. For model training purposes, the Scikit-learn (sklearn) library will be leveraged. The Scikit-learn library provides an extensive collection of machine learning algorithms, comprising supervised, semi-supervised, unsupervised, and reinforcement learning algorithms. These algorithms empower researchers to explore diverse methodologies in addressing the intricacies of the datasets and enhancing the robustness of the models developed throughout the study.

In our study, we aim to develop a real-time web application intended for users to detect attacks on WSNs and discern the specific type of attack occurring during data packet transmission between nodes. To accomplish this, we employ a combination of technologies including HTML, Bootstrap, JavaScript, Pickle, Django, and SQLite. HTML serves as the foundation for structuring the content of the web application, providing a framework for presenting information to users. Bootstrap is utilized to enhance the visual appeal of the application, offering styling elements that contribute to a visually engaging user experience. JavaScript facilitates interactive features within the web application, enabling seamless communication and interaction between users and the application interface. Through JavaScript, users can input data and receive responses from the application in real-time. Django serves as the primary backend framework for the web application. Its functionality encompasses data handling, communication with the machine learning model, and response generation. Django effectively manages the flow of data from users to the machine learning model and back, ensuring smooth operation and efficient processing.

Pickle is employed for serializing and deserializing Python objects, enabling the storage and retrieval of machine learning models within the application. This allows for seamless integration of the trained models into the Django backend.

Additionally, SQLite is utilized as the database management system, providing a lightweight and efficient solution for storing and retrieving data within the web application.

Together, these technologies enable the development of a robust and user-friendly web application capable of detecting and identifying WSN attacks in real-time, thereby enhancing the security of WSNs in practical scenarios **(Figure 1)**.

System architecture

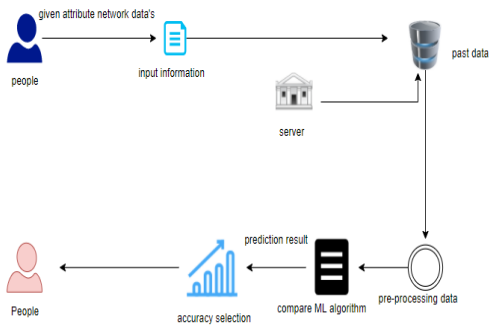

Figure 1. Architecture of our system.

Dataset

The dataset utilized in this proposal was developed by Almomani et al. This dataset encompasses 18 features and 5 classes, aimed at detecting four types of attacks: Black hole attack, gray hole attack, flooding, and TDMA.

It comprises over 374,662 records. This dataset for WSNs will be accessible on the Kaggle website. Kaggle serves as a platform for data science competitions, where participants engage in creating optimal models for addressing specific problems or analyzing particular datasets. However, our dataset is characterized by a lack of balance among the five classes. Each class within the dataset does not possess an equal number of records, thus posing challenges to the accuracy of machine learning models **(Table 1 and Figure 2)**.

Table 1. WSN–features.

Feature No.	Feature name
1	ADV-S
2	Is-CH
3	Rank
4	SCH-S
5	Send-code
6	JOIN-S
7	Dist-To-CH
8	SCH-R
9	JOIN-R
10	DATA-S
11	Data-sent-To-BS
12	AVD-R
13	Who-CH
14	ID
15	Expaned Energy
16	Time
17	Dist_CH_To_BS
18	DATA-R

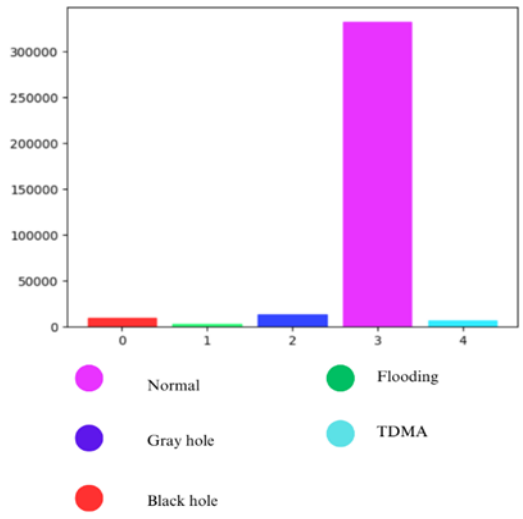

Figure 2. Imbalance dataset.

The imbalance within our dataset manifests with a significant disparity in the number of records allocated to the "normal" class compared to the other classes. This imbalance poses a substantial risk of biasing our machine learning model. Consequently, the accuracy of our model

in detecting WSN attacks is compromised, leading to potential failures within the machine learning framework.

To rectify the imbalance within our dataset, we will implement the oversampling technique using imblearn, a specialized library tailored to address imbalanced datasets. Imblearn offers a range of methodologies including under sampling, oversampling, and SMOTE (Synthetic Minority Over-Sampling Technique) to mitigate and rectify the imbalance inherent in datasets. By leveraging these techniques, we can effectively balance our dataset, thereby enhancing the representativeness of minority classes. With a balanced dataset at our disposal, we can proceed to train our machine learning model with greater confidence, anticipating higher accuracy in the detection of WSN attacks **(Figure 3)**.

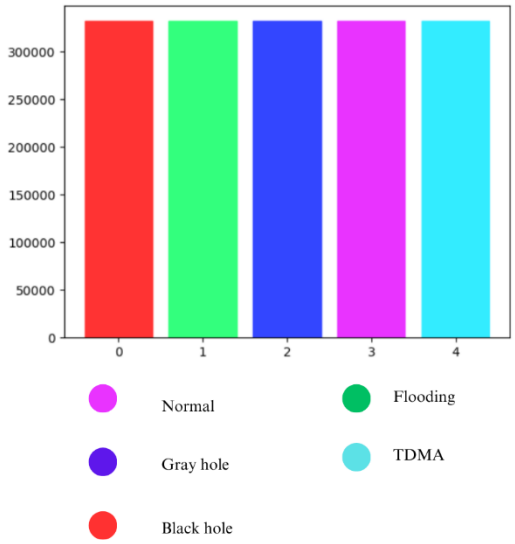

Figure 3. Balanced dataset.

Feature selection

In this study, we have opted to incorporate all 18 features into our training regimen for the classifiers. This approach is aimed at fortifying the robustness of our machine learning model. By including all features in the training process, we anticipate a notable enhancement in the performance of the classifiers. The rationale behind this decision lies in the belief that each feature contributes unique insights and predictive power to the model. By leveraging the entirety of available features, we seek to maximize the discriminative capability of the classifiers, thereby bolstering their overall performance in accurately detecting and classifying WSN attacks.

Classifiers

Machine learning assumes a paramount role in fortifying the security infrastructure of Wireless Sensor Networks (WSNs), as it offers a sophisticated mechanism for detecting potential threats within the network. Through the application of machine learning techniques, the WSNs can effectively discern and mitigate various forms of attacks targeting their integrity and functionality. Within our study, we intend to employ three distinct classifiers aimed at detecting potential attacks within Wireless Sensor Networks (WSNs). Each of these classifiers will exhibit unique characteristics and performance metrics, which we will evaluate primarily based on their accuracy scores. The classifiers demonstrating high accuracy scores will be selected to form the foundation of our WSN attack prevention methodology **(Figure 4).**

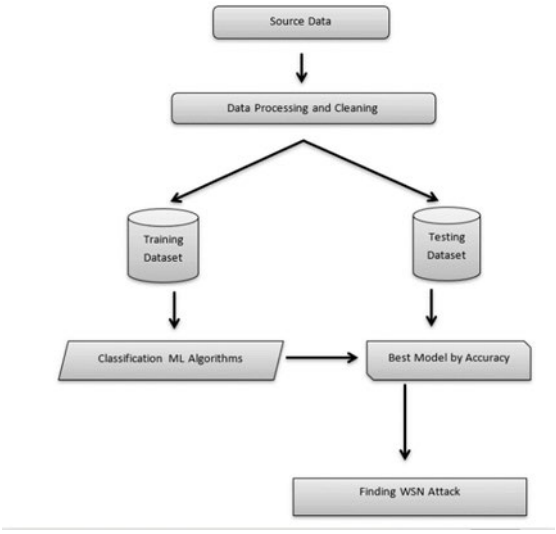

Figure 4. Workflow diagram.

Classifiers

- BernoulinNB
- Ridge classifier
- Randomforest classifier

Based on the ideas of the Naive Bayes theorem, the Bernoulli Naive Bayes algorithm (BernoulliNB) is a classification algorithm. It is especially made for situations involving binary classification, in which there are two alternative classes for the target variable. For each data point, the predicted class is the one with the highest posterior probability. When using binary classification, the predicted class is the one with the higher probability (for example, class 1 is predicted if P(Y=1|X)>P(Y=0|X).

For classification tasks, a machine learning technique called the ridge classifier is employed. It is a variation of the method known as Ridge Regression, which is mainly applied to regression applications. For multi-class classification situations, where the objective is to allocate an input data point to one of multiple predetermined classes or categories, the Ridge Classifier was created expressly. "Alpha" (α), a hyper parameter of the ridge classifier, regulates how strong the L2 regularization is. More alpha will result in greater regularization and, in turn, more straightforward models. Methods like cross-validation are usually used to establish the proper value of alpha. Multiple class classification jobs are naturally handled by the Ridge Classifier. To expand binary classification to several classes, it usually uses a One-*vs*-One (OvO) or One-*vs*-Rest (OvR) technique.

A potent machine learning approach for both classification and regression applications is the Random Forest Classifier. It is an ensemble learning technique that creates a more reliable and accurate model by combining the predictions of several decision trees. To maximize their effectiveness, Random Forests can be adjusted for a number of hyper parameters, such as: how many trees there are in the forest. The deepest that any tree can go. The bare minimum of samples needed to divide a node. The most features that can be taken into account at each split. The standard by which split quality is evaluated (e.g., entropy for classification or Gini impurity).

Validation

In our study, all experiments will undergo rigorous validation processes. The validation metrics encompass accuracy, precision, recall, and F1 score. Based on the scores derived from these metrics, we will conduct evaluations to determine the machine learning model that attains the highest overall performance. Subsequently, the selected model will be utilized to enhance the security framework of Wireless Sensor Networks (WSNs).

Accuracy: The percentage of all predictions that are accurate; that is, the overall frequency with which the model accurately predicts defaulters and non-defaulters. The easiest performance metric to understand is accuracy, which is just the ratio of properly predicted observations to total observations. If our accuracy is high, one might assume that our model is the best. Indeed, accuracy is an excellent statistic, but only in cases when the datasets are symmetric, meaning that the false positive and false negative values are almost equal.

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

False Positive (FP): Someone who pays as expected but is not in default. when the intended class is yes but the real class is no. For example, suppose the actual class states that the passenger did not survive whereas the forecast class predicts that they will.

False Negatives (FN): A non-paying individual who is expected to default. When the anticipated class is not the actual class. For example, if the predicted class informs you that the passenger will die and the actual class result

shows that this passenger survived.

True Positives (TP): A nonpaying individual who is anticipated to be in default. These are the positively predicted values that were accurate, indicating that both the anticipated and actual values for the class are yes. For example, if the projected class and actual class values both suggest that this passenger survived, then you know the same thing.

True Negatives (TN): An individual who is expected to default on payments. These are the accurately predicted negative values, indicating that both the actual and anticipated values for the class are zero. For example, if the anticipated class reports the same thing and the actual class reports that the passenger did not survive.

Precision refers to the percentage of optimistic projections that come true. Precision is defined as the ratio of precisely anticipated positive observations to all predicted positive observations. How many passengers who were reported to have survived actually did? is the question that this metric tries to answer. A low false positive rate correlates with high accuracy.

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

Recall: The percentage of observed positive values that were accurately anticipated. (The percentage of real defaulters that the model can accurately forecast). The ratio of accurately predicted positive observations to all observations made in the actual class is known as recall.

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

F1 score: The Precision and Recall weighted average is known as the F1 score. Even though it is harder to understand intuitively, F1 is typically more essential than accuracy, especially when classes are unevenly distributed. Accuracy performs best when the costs of false positives and false negatives are comparable. If the costs of false positives and false negatives differ significantly, it is preferable to include both precision and recall.

$$F1Score = 2X\frac{PrecisionXRecall}{Precision + Recall}$$

Web application

In our study, we have embarked on the development of a real-time web application. This application is designed to serve as a robust tool for detecting potential attacks within Wireless Sensor Networks (WSNs) during the transmission of data packets between nodes. The architecture of our web application is structured into three primary modules, each comprising numerous sub-modules aimed at augmenting the functionality and features of our application. By leveraging our web application, users gain the capability to identify and classify attacks targeting WSNs. This is facilitated through an intuitive user interface, where individuals can input relevant parameters based on the features extracted from the dataset utilized to train our model. Through this interactive process, users can effectively discern the presence of attacks within WSNs and ascertain the specific type of attack occurring within the network environment **(Figure 5).**

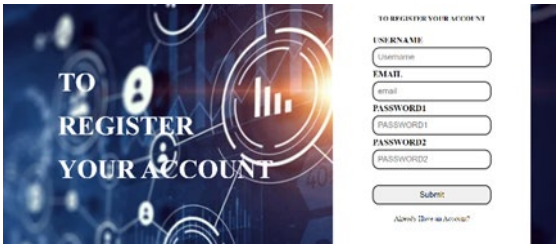

Figure 5. User registration.

User registration: Within this module, users are required to register using their email addresses and designate a password to gain access to our web application. However, if a user holds the status of administrator on our website, the registration process is unnecessary. Administrators can simply log in directly using their email addresses and passwords, bypassing the registration step **(Figure 6)**.

Figure 6. User login.

User login: In this module, if a user has already registered on our website, there is no requirement for re-registration. Instead, users can directly log in using their email addresses and passwords. However, it is imperative

that the provided email address and password match the credentials stored in our database for authentication purposes. Our website employs stringent security measures to ensure that only legitimate users are granted access. Malicious attempts to access the website are actively thwarted, and only authenticated users with valid credentials are permitted entry **(Figure 7)**.

Figure 7. Prediction.

Prediction: In this module, users will furnish all the requisite values corresponding to the features outlined within the module. These values serve as inputs representing the features of the dataset utilized for training our model. Upon user submission of the input values, the data is transmitted to our machine learning model using the Django technology. Django facilitates the seamless transmission of data from the web application to the machine learning model. Subsequently, the model processes the input data and generates an output based on its trained parameters and algorithms. This output is then returned to the user as the response from our trained machine learning module.

RESULTS AND DISSCUSSIONS

In this section, we shall engage in a comprehensive discourse concerning the outcomes derived from our experimental endeavor involving the training of our machine learning model using the Wireless Sensor Network (WSN) dataset. Furthermore, we shall delve into a detailed examination of the merits and demerits inherent in each model. Additionally, we will undertake a thorough comparison of these machine learning models, focusing on a spectrum of performance metrics, inclusive of accuracy, precision, recall, and F1 score. Moreover, we endeavor to expound upon these metrics through the elaborate utilization of a confusion matrix, which affords a comprehensive visualization of the models' predictive capabilities and error patterns.

From the preceding performance analysis figure, we can glean insights into the performance of each classifier trained using the WSN dataset. It is evident that the random forest classifier exhibits superior accuracy in detecting WSN attacks compared to both the ridge classifier and bernoullinb classifier **(Figure 8)**.

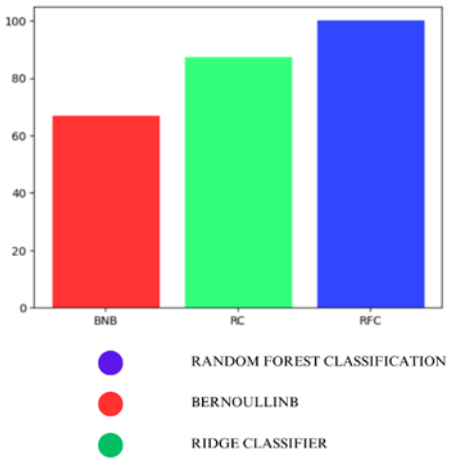

Figure 8. Performance analysis of classifiers.

Now, let us proceed to examine the confusion matrix corresponding to each classifier utilized in our experimental analysis.

The Bernoulli Naive Bayes classifier yields an accuracy rate of 66.86%. Additionally, it furnishes metric values such as precision, registering at 71%, recall at 67%, and

an f1-score of 61%. Notably, among all algorithms engaged in the experiment, BernoulliNB demonstrates the least performance **(Figure 9)**.

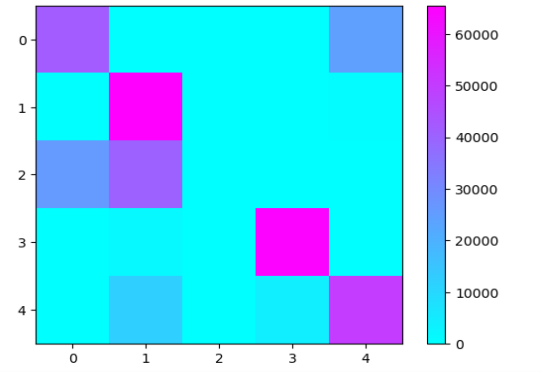

Figure 9. Confusion matrix of BernoulinNB.

The confusion matrix linked with the Bernoulli Naive Bayes classifier provides a thorough comprehension of precision, recall, and F1 score. Precision is defined as the proportion of accurately predicted positive observations to all expected positives. Recall, also known as sensitivity, is the proportion of correctly predicted positive observations to all observations in the class. The F1 score is the harmonic mean of precision and recall, providing a fair evaluation of the model's effectiveness **(Figure 10)**.

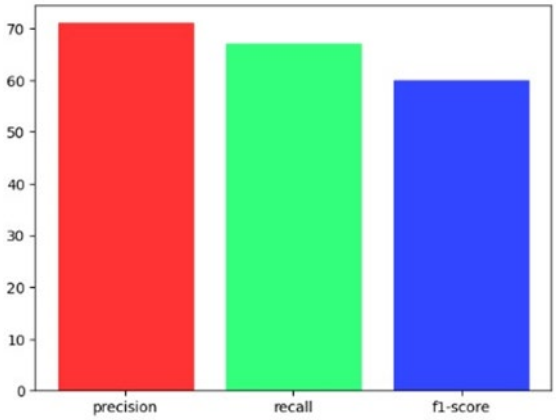

Figure 10. Graphical representation.

The ridge classifier yields an accuracy rate of 87%. Moreover, it furnishes metric values such as precision, registering at 89, recall at 87, and an f1-score of 87. Notably, among all algorithms engaged in the experiment, the ridge classifier exhibits moderate performance **(Figures 11-12)**.

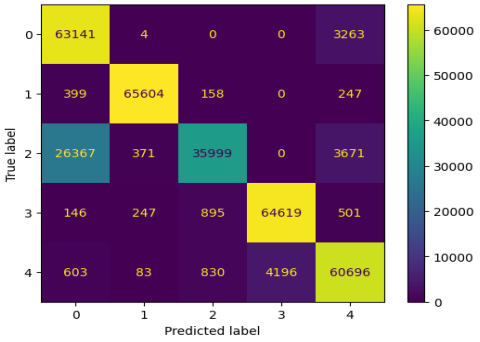

Figure 11. Confusion matrix of ridge classifier.

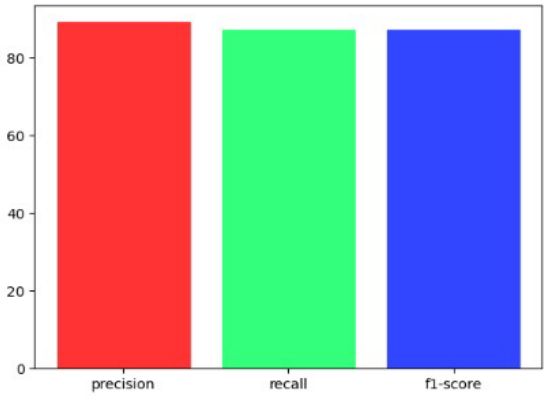

Figure 12. Graphical representation.

The random forest classifier delivers an accuracy rate of 99%. Additionally, it presents metric values such as precision, achieving 100, recall with 100, and an f1-score of 100. Remarkably, among all algorithms incorporated in the experiment, the random forest classifier demonstrates the highest performance **(Figure 13-14)**.

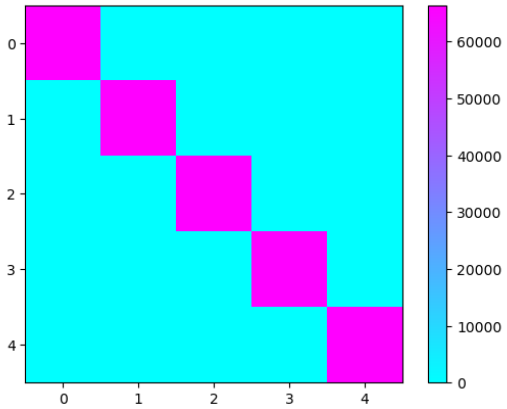

Figure 13. Confusion matrix of random forest classifier.

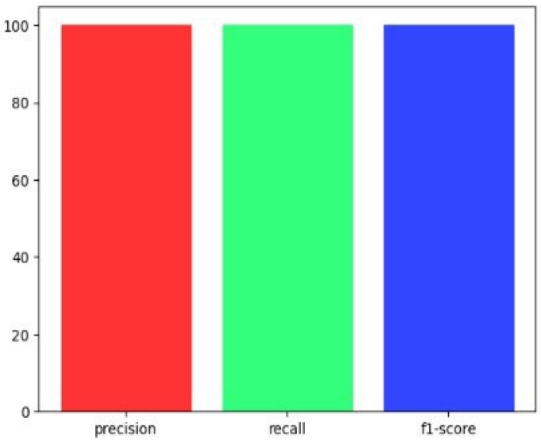

Figure 14. Graphical representation.

CONCLUSION

Wireless Sensor Networks (WSNs) have surfaced as one of the most promising options for a range of applications, such as distributed control systems, healthcare, defense, and environmental monitoring. In a WSN, smart sensor nodes are affordable and easy to configure. Nevertheless, due to their placement in hostile environments, unsecure routing methods, and unsound architectural designs, making them vulnerable to a variety of threats.

Our study concludes that through the training of three distinct machine learning models utilizing the Wireless

Sensor Network (WSN) dataset encompassing over 300,000 records, we aim to predict attacks during data transmission between nodes and identify the specific type of attack. We conducted a comparative analysis among the three machine learning models. The Bernoulli Naive Bayes classifier exhibited an accuracy rate of 66%, while the ridge classifier achieved an accuracy rate of 87%. However, the random forest classifier surpassed both with an impressive accuracy rate of 99%. Consequently, we have decided to deploy the random forest classifier in a real-time web application for user access, leveraging its superior accuracy and performance.

REFERENCES

1. Salmi S, Oughdir L (2023). Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network. J Big Data. 10(1):17.
2. Wazirali R, Ahmad R (2022). Machine learning approaches to Detect DoS and their effect on WSNs Lifetime. Comput Mater Continua. 70(3).
3. Ismail S, Dawoud D, Reza H (2022). A lightweight multilayer machine learning detection system for cyber-attacks in WSN. IEEE 12th annual computing and communication workshop and conference (CCWC). 0481-0486.
4. Ifzarne S, Tabbaa H, Hafidi I, Lamghari N (2021). Anomaly detection machine learning techniques in wireless sensor networks. J Phys Conf. 1743(1).
5. Alsulaiman L, Al-Ahmadi S (2021). Performance evaluation of machine learning techniques for DOS detection in wireless sensor network. arXiv. 2104:01963.
6. Tabbaa H, Ifzarne S, Hafidi I (2022). An online ensemble learning model for detecting attacks in wireless sensor networks. arXiv. 2204:13814.
7. Gebremariam GG, Panda J, Indu S (2023). Localization and detection of multiple attacks in wireless sensor networks using artificial neural network. Wirel Commun Mob Comput. 2023(1):2744706.
8. Alsahli MS, Almasri MM, Al-Akhras M, Al-Issa AI, Alawairdhi M (2021). Evaluation of machine learning algorithms for intrusion detection system in WSN. Int J Adv Comput Sci Appl. 12(5).
9. Baraneetharan E (2020). Role of machine learning algorithms intrusion detection in WSNs: A survey. J Inf Technol Digit World. 2(3):161-173.
10. Almomani I, Al-Kasasbeh B, Al-Akhras M (2016). WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. J Sens. 2016(1):4731953.
11. Ashraf S, Alfandi O, Ahmad A, Khattak AM, Hayat B, et al (2020). Bodacious-instance coverage mechanism for wireless sensor network. Wirel Commun Mob Comput. 2020(1):8833767.
12. Feng X, Ding X, Sun S (2013). A security detection scheme based on evidence nodes in wireless sensor networks. in Proc 6th Int Conf Biomed Eng Informat. 689–693.
13. Kumar BS, Sinha S (2022). An Intrusion Detection and Prevention System against DOS Attacks for Internet-Integrated WSN. In Proc 7th Int Conf Commun Electron Syst (ICCES). 793-797.
14. Rao GS, Harshitha M, Joshitha VR, Sravya SS, Priya MV (2023). DoS Attack Detection in Wireless Sensor Networks (WSN) Using Hybrid Machine Learning Model. In Proc 10th Int Conf Signal Process Integr Netw (SPIN). 384-388.
15. Tomic I, McCann JA (2017). A survey of potential security issues in existing wireless sensor network protocols. IEEE Internet Things J. 4(6):1910-1923.
16. Shi E, Perrig A (2004). Designing secure sensor networks. IEEE Wirel Commun. 11(6):38-43.
17. Premkumar M, Ashokkumar SR, Jeevanantham V, Mohanbabu G, AnuPallavi S (2023). Scalable and energy efficient cluster based anomaly detection against denial of service attacks in wireless sensor networks. Wirel Pers Commun. 129(4):2669-2691.
18. Pan JS, Fan F, Chu SC, Zhao HQ, Liu GY (2021). A light weight intelligent intrusion detection model for wireless sensor networks. Secur Commun Netw. 2021(1):5540895.
19. Dener M, Al S, Orman A (2022). STLGBM-DDS: An efficient data balanced DoS detection system for wireless sensor networks on big data environment. IEEE Access. 10:931-945.
20. Roman R, Lopez J (2009). Integrating wireless sensor networks and the internet: A security analysis. Internet Res. 19(2):246-59.