International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Machine Learning Algorithms to Detect Attacks in Wireless Sensor Networks

Neha Jagwani*[1], Dr. Poornima G.*[2]

Submitted: 29/11/2023 **Revised**: 09/01/2024 **Accepted**: 19/01/2024

Abstract: A network of independent and interconnected sensor nodes that communicate with each other wirelessly to collect, process, and transmit data from the environment they are deployed in is referred to as Wireless Sensor Network. These nodes are equipped with varied types of sensors, such as temperature, humidity, light, motion, and gas, enabling them to monitor and gather information about their surroundings. The data collected by these nodes can be utilized for various applications, making WSNs an integral part of modern technological advancements. Wireless Sensor Networks are subject to various types of attacks due to their inherent characteristics, limited resources, dynamic topologies, and wireless communication. These attacks can compromise the network's integrity, confidentiality, availability, and overall functionality. In this paper, we have focused on DOS, Probe, R2L and U2R attacks. Many Machine Learning algorithms have been applied to detect these attacks in WSNs. ML algorithms have also been compared after applying a balancing technique called SMOTE. Binary and multi-class classifications have been performed to detect the attacks in WSN. The algorithms are compared based on performance matrices like MCC, CS, ROC, weighted and macro average scores, recall precision, and F-1 scores.

Keywords: *Wireless Sensor Networks, Sensor Nodes, Denial of Service, Probing Attack, Remote to Local, User to Root, Machine Learning*

1. Introduction

The Wireless Sensor Networks (WSNs) consist of thousands of Sensor Nodes (SNs) connected through wireless communication. These SNs are small-sized and low-powered devices comprised of a power unit, computing and processing unit, sensing unit and transceiver. The nodes sense the environment/ physical quantity and gather the information that can be transmitted directly or through multiple hops to sink, which can then be used locally or transferred to other networks through gateways. Researchers see the Sensor Network systems as significant technology that will experience extensive deployment for abundant application in the next few years. Some applications include factory and industrial monitoring, environmental and animal monitoring, health monitoring, agricultural monitoring, and automation.

Based on the application, sometimes SNs are positioned in distant and hostile locations where human intervention is impossible. Moreover, sensor nodes are limited in power, memory and computational capacities. Thus, they are inclined to failure because of limited resources and the environment in which they are installed. It is necessary to make the WSNs fault-free due to their role in many applications.

[1,2]*Department of Electronics and Communication Engineering*
[1] *Research Scholar, BMS College of Engineering, Bangalore, India*
[2] *Associate Professor, BMS College of Engineering, Bangalore, India*
E-mail Id: [1] jagwanineha@gmail.com, [2] gpoornima.ece@bmsce.ac.in
** Corresponding Author: Neha Jagwani*
Email: jagwanineha@gmail.com

Security is one of the major challenges in WSN due to factors including low transmission range, wireless medium, hostile environment, Ad hoc deployment and limited energy.

One of the most promising technologies that can keep the WSNs fault-free is using Deep Learning and Machine Learning techniques. In this paper, we have concentrated on the Improvement of software faults in WSN with the support of Deep Learning and Machine Learning Techniques. This paper is structured as various types of attacks in Sensor Networks using different ML techniques followed by detecting various attacks in WSN using ML algorithms.

1.1 Types of attacks in WSNs

WSNs have extremely constrained computational resources compared to other sensory equipment and are comparably inexpensive. However, their limited processing power and computational capacity make wireless sensor nodes susceptible to security threats. As we know, WSNs are left unattended for a long time and mostly used in remote areas, so they provide an easy target for physical attacks and unauthorized access. A few types of attacks are considered in this paper, which are explained below:

1. **DOS (Denial of Service) attack:** These types of attacks block and hinder services delivered to the user by the network.
2. **Probe attack**: These attacks aim to attain information about the network or computer system.
3. **U2R (User to Root) attack**: These types of attacks aim to gain root or admin-user access by a non-privileged

user on a specific computer or system on which the attacker had user-level access. These attacks obtain root privileges illegally while accessing a local machine.

4. **R2L (Remote to Local) attack**: The intruder gains unauthorized access to a victim machine in the entire network.

It is important to prevent WSNs from these attacks as they are used in various critical applications. In this paper, we have used machine learning algorithms to detect these attacks.

2. Literature Review

Elsadig et al. (2023), The rapid expansion of wireless sensor networks (WSNs) in various fields is due to their unique characteristics and performance. However, WSNs are highly vulnerable to security attacks, especially denial-of-service (DoS) attacks. This study focuses on identifying WSN limitations, weaknesses, and security threats, specifically focusing on DoS attacks. It explores recent techniques for detecting DoS attacks and proposes a lightweight machine learning approach using a decision tree (DT) algorithm with the Gini feature selection method. The approach achieves a 99.5% accuracy rate with minimal processing overhead compared to other classifiers. [1]

Lai et al. (2023), WSNs face security challenges that can drain their limited energy resources. Traditional security protocols are inadequate due to communication and resource constraints. This study proposes an online learning-based approach to detect denial-of-service (DoS) attacks in WSNs. It introduces a feature selection method and a noise-tolerant online passive-aggressive multi-class classifier. The method is evaluated based on accuracy, precision, recall, and F1-score, demonstrating competitive performance. [2]

Ahmad et al. (2022), Wireless sensor networks (WSNs) encounter significant challenges related to energy and security. As security complexity increases, energy consumption also rises. Traditional security protocols are not effective in WSNs due to resource limitations. This paper discusses the potential of machine learning algorithms in enhancing WSN security while reducing costs. It addresses challenges and solutions for sensors to identify threats, attacks, and malicious nodes through machine learning. Open issues related to adapting machine learning algorithms to WSN capabilities are also explored. [3]

Salmi et al. (2023), Wireless Sensor Networks (WSNs) are susceptible to various security threats, especially denial-of-service (DoS) attacks. Traditional intrusion detection systems are becoming less effective against intelligent and complex attacks. This study reviews related works on DoS attack detection in WSNs and develops deep learning-based intrusion detection systems trained on a specialized dataset. These systems are evaluated and compared, focusing on four types of DoS attacks: Blackhole, Grayhole, Flooding, and Scheduling. [4]

Diener et al. (2023), The popularity of Wireless Sensor Networks (WSNs) has attracted attackers who aim to infiltrate, capture, and manipulate these networks. Layers categorize attacks in WSNs, and network traffic data is analyzed to prevent future attacks. Learning models are used to preprocess and categorize network data, leading to more accurate attack detection than traditional methods. This study focuses on WSN network layer attacks. It presents results from machine learning and deep learning models, including Random Forest, Decision Tree, Naive Bayes, Logistic Regression, MLP, CNN, LSTM, GRU, CNN-LSTM, LSTM-CNN, CNN-GRU, and GRU-CNN, using the WSN-BFSF dataset. [5]

Gebremariam et al. (2023), Research in wireless sensor networks (WSNs) focuses on identifying and localizing malicious nodes, which can extend the network's lifespan and enhance its value. Anchor nodes with known positions are used to estimate the positions of unknown nodes. Various localization methods exist for precise node estimation. However, setting suitable network parameters for accurate node localization during network setup remains challenging. Routing attacks, such as wormhole attacks, Sybil attacks, blackhole attacks, and replay attacks, can impact localization accuracy and the quality of service in WSNs. This work proposes a secure localization and routing threat detection approach in WSNs using optimized hybrid machine learning methods for optimization of distance, position, and data communication. The approach utilizes benchmark datasets CICIDS2017 and UNSW NB15 to calculate average localization accuracy and identify malicious nodes. The system achieves a 100% average detection accuracy and significantly improves localization accuracy, with an average error of 0.191. [6]

Ismail et al. (2022), Wireless Sensor Networks (WSNs) are crucial for the Internet of Things (IoT) but are energy-constrained. The increase in deployed sensors has made security a major concern, requiring effective detection and mitigation methods. This paper presents the Weighted Score Selector (WSS), a lightweight ensemble-based machine-learning approach for cyber-attack detection in WSNs. WSS combines supervised ML classifiers dynamically to improve detection performance quickly. It outperforms classical ensemble techniques in terms of various metrics. The approach is evaluated using the WSN-DS dataset, demonstrating promising results in detecting Denial of Service attacks. [7]

Abidoye et al. (2023), Wireless sensor networks (WSNs) are innovative but vulnerable to various attacks, including denial of service (DoS) attacks. Designing effective detection and prevention systems for WSNs is challenging. This study suggests using machine learning models,

specifically decision trees and XGBoost, to identify DoS attacks in WSNs. Extensive tests on WSN datasets show that XGBoost outperforms decision trees in terms of true positive rates and false positive rates, demonstrating its effectiveness in detecting DoS attacks. [8]

Due to their unique characteristics and limitations, Ismail et al. (2023) and Wireless Sensor Networks (WSNs) face cybersecurity challenges. This paper provides a comprehensive overview of cybersecurity principles in WSNs and explores current and envisioned solutions, focusing on Machine Learning (ML) and Blockchain (BC) security techniques. It discusses integrating BC and ML to develop a lightweight security framework for cyberattack detection and prevention in WSNs, emphasizing design insights and challenges. The paper proposes an integrated BC and ML solution for WSN security. [9]

Mounica et al. (2021), Wireless Sensor Networks (WSNs) play a critical role in military and civilian applications, particularly in sensitive areas like battlefields. Developing security measures for these networks is of utmost importance to enhance their reliability and quality of service. However, deploying WSNs for security exposes them to various viruses and hacking threats. One significant threat is the Sybil Attack, where malicious nodes impersonate multiple false identities simultaneously, deceiving legitimate nodes. A machine learning model is proposed to detect Sybil attacks by analyzing raw traffic data and distinguishing between authorized and unauthorized access points (APs) in a combined wired and wireless environment to address this. [10]

Gebremariam et al. (2023), Security enhancement in wireless sensor networks (WSNs) is crucial, especially against routing attacks that inject malicious nodes. Sybil attacks are common routing attacks where false nodes are generated. This paper introduces a detection and localization scheme using an optimized multilayer perceptron artificial neural network (MLPANN) to combat various attacks, including Denial of Service (DoS) attacks. The system is implemented and evaluated using benchmark datasets, achieving high detection accuracy and precise localization, making it suitable for scalable and hierarchically distributed WSNs. [11]

Kousar et al. Wireless Sensor Networks (WSNs) are widely used but are vulnerable to security threats, including sinkhole attacks, where malicious nodes redirect data from sensor nodes. Existing techniques for sinkhole attack detection suffer from high false alarm rates, leading to low accuracy and excessive energy consumption. This article reviews the literature on sinkhole attack detection in WSNs and proposes a Sinkhole Attack Detection with Machine Learning (SAD_ML) technique. The method uses AODV protocol and machine learning algorithms, particularly SVM, to achieve a high detection accuracy of 96%. [12]

Ifzarne et al. (2021) , Wireless Sensor Networks (WSNs) have grown rapidly in various applications, but they face security challenges due to their deployment in unattended and hostile environments. This work focuses on intrusion detection in WSNs, specifically using online learning classifiers. The proposed model combines information gain ratio feature selection with an online Passive-aggressive classifier. Experiments on the WSN-DS dataset demonstrate the model's effectiveness in detecting various attacks, achieving a 96% detection rate, with 86% accuracy for scheduling attacks and 99% for normal traffic. [13]

Saleh et al. (2024) , Wireless Sensor Networks (WSNs) are integral to cyber-physical systems but are susceptible to cyberattacks. This Research employs machine learning techniques, such as Gaussian Nave Bayes (GNB) and Stochastic Gradient Descent (SGD) algorithms, to enhance intrusion detection in WSNs. Principal component analysis and singular value decomposition are applied to raw traffic data to reduce the computational burden. The proposed SG-IDS model achieves a 96% accuracy rate on the WSN-DS dataset, outperforming state-of-the-art algorithms in intrusion detection tasks. It also demonstrates excellent performance in an Internet of Medical Things (IoMT) dataset evaluation, further validating its effectiveness. [14]

Yu et al. (2021) , Wireless Sensor Networks (WSNs) are crucial in various practical applications, including sensitive areas like battlefields. However, security is a significant concern due to the risk of denial-of-service (DoS) attacks. This Research aims to enhance WSN security by detecting and responding to DoS attacks. Two types of machine learning techniques, neural networks (NN) and Support Vector Machines (SVM), are used to detect attacks at the media access control (MAC) layer. The study compares the effectiveness of these two methods in a wireless sensor node's access channel. Simulations are conducted using a wireless network simulator, Vanderbilt plow error simulation. [15]

Annapurna et al. (2023) , Wireless Sensor Networks (WSNs) are widely used in monitoring applications but face security threats, particularly from malicious nodes causing security attacks. These attacks occur in different network layers and can lead to unauthorized data access. Wormhole and Sybil attacks are examples of such threats. To prevent further damage, a prediction module with various machine learning algorithms, including XGBoost, Adaboost, Random Forest, and KNN, is used to classify attacks based on a dataset called WHASA. The goal is to classify network access as "normal" or "under attack" due to Wormhole and Sybil attacks. [16]

Gebremariam et al. (2023), Internet of Things Wireless Sensor Networks (IoT-WSNs) are essential for various applications, including healthcare and security. However, security threats can arise from multiple sources. This study proposes a secure attack localization and detection method in IoT-WSNs using blockchain-based cascade encryption and trust evaluation. This approach rewards nodes for service provisioning and trust, removing malicious nodes that degrade localization accuracy and service quality. Federated machine learning combines various techniques to classify harmful nodes through feature assessment. The proposed system achieves high detection and classification accuracy, making it suitable for large-scale IoT-WSNs. [17]

Bagwari et al. (2023), Industrial Wireless Sensor Networks (WSNs) offer scalability and cost advantages but pose energy optimization and maintenance challenges. Machine Learning techniques are employed to create an enhanced energy optimization model for Industrial WSNs. This model identifies and optimizes node energy consumption, evaluates feedback control schemes, and predicts optimal outcomes. It also explores trade-offs between power consumption and communication performance. The proposed model demonstrates significant energy savings and improved network efficiency, making Industrial WSNs more reliable **and cost-effective, particularly when combined with manual intervention. [18]**

Oleiwi et al. (2023), The rapid advancement of wireless communication systems, particularly in the context of 6G and beyond, has introduced new features and challenges. Machine learning techniques have gained prominence in various fields, including wireless communications, focusing on improving network traffic performance in resource management, frequency spectrum optimization, latency, and security. However, the evolving landscape of modern wireless communications has revealed vulnerabilities, necessitating the development of a robust intrusion detection system (IDS). Existing IDSs have struggled to provide adequate protection against sophisticated attacks in 6G networks, resulting in low accuracy and high false alarm rates. This paper presents a meta-machine learning model for anomaly detection in wireless communication networks to address this challenge. The proposed approach involves dataset accumulation, preprocessing, feature selection, utilization of various classifiers, and a meta-model classifier for enhanced intrusion detection. Experimental results demonstrate the high efficiency and superiority of the proposed approach compared to existing IDSs. [19]

Abhale et al. (2023), concerns about unauthorized data access have grown with the increasing use of various communication protocols. Advanced intrusion detection systems (IDSs) have been developed to address these security issues. Deep learning, a type of machine learning, has gained attention due to its success in various domains and the availability of large-scale datasets. Researchers have turned to deep learning to enhance intrusion detection capabilities in network security. This paper reviews the current state of deep learning-based IDSs and compares them with modified algorithms proposed in recent Research. [20]

Alqahtani et al. (2019), Intrusion detection systems (IDSs) play a critical role in safeguarding the services and infrastructures of wireless sensor networks (WSNs) from unforeseen attacks. While machine learning-based IDSs have shown promise, they still face challenges in achieving accuracy and efficiency, especially when dealing with imbalanced network traffic data. This paper introduces a novel model, the GXGBoost model, for intrusion detection in WSNs. This model combines genetic algorithms with an extreme gradient boosting (XGBoost) classifier to improve the detection of minority-class attacks in highly imbalanced WSN traffic data. Extensive experiments on the WSN-DS dataset demonstrate that the proposed approach outperforms state-of-the-art methods and achieves high detection rates for various types of attacks, including flooding, scheduling, grayhole, and blackhole attacks, as well as normal traffic. [21]

3. Machine learning algorithms

In machine learning, computers learn with their experience without human intervention and reprogramming. The process starts with providing good-quality data to the machine and then training them to build an ML model using the given data and different algorithms. This is applied to test data to check the model efficiency, which is then used for future predictions. Fig 1 shows the flow chart of the machine learning algorithm.

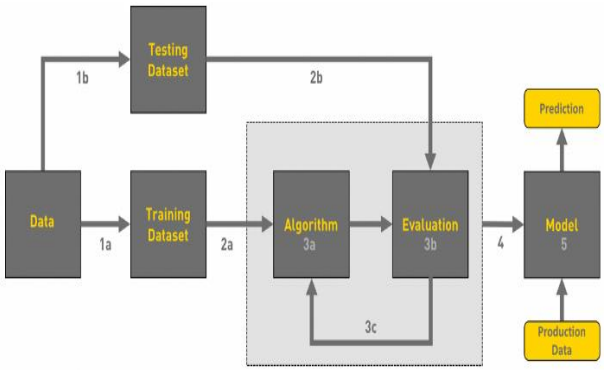

Fig 1. Flow chart of Machine Learning Model

Three types of Machine Learning Algorithms are:

1. Supervised Machine Learning: It can be considered as learning, guided by a trainer/Teacher, where the dataset acts as a trainer that trains the model so that whenever new data is provided to the model, it can make the decision. Thus, it makes predictions based on historical data. Supervised machine learning can be classified as:

- Classification: When the output or target variable is a category. It can be yes or no, positive or negative, 0 or 1 etc.
- Regression: When output is a continuous value, i.e., it is used to predict the numerical data instead of labels.

2. Unsupervised Machine Learning: The model finds structures in the dataset and groups the data into clusters according to the observed relationships and patterns.
3. Reinforcement Machine Learning: It is a trial-and-error method in which the agent is rewarded for the desired output and penalized for the undesired one. Then, based on positive reward points achieved by the agent, the model trains itself and prepares to make decisions for the new data.

Below is a brief description of a few Supervised Machine Learning algorithms mentioned in the paper.

1. **Logistic Regression**: It is used when the target variable is the binary classification of data points. It predicts P(y=1) as a function of x. A 'S' shaped curve known as the logistic or sigmoid function changes the real values between 0 and 1. Y is classified in the 0 category if the output of this logistic function is less than 0.5 or the graph goes to a negative end. Conversely, if the output exceeds 0.5 or the graph goes to the positive side, y is predicted as 1.
2. **Decision Trees (DT) and Random Forest (RF)**: Decision trees are used for classification and regression problems. The algorithm divides the dataset into branches and again into other branches until a leaf node is accomplished. The root node or the topmost node of the decision tree is responsible for splitting the dataset using the feature that results in the best split. At times, a single tree is not enough to generate effective results. This is where the RF algorithm comes into play. It is an ensemble learning method where multiple decision trees combine to determine the outcome instead of depending on a single tree. Each decision tree is trained on a different data sample, and every tree uses a subset of features selected at the node's splitting point.
3. **Support Vector Machines (SVM) are applied to** classification and regression problems. This algorithm creates the best line (also called decision boundary) which can segregate n-dimensional space into distinct classes so that new data point can be placed into the correct class/category in the future.
4. **Naive Bayes (NB)**: It is built on the Bayes theorem, which assumes that the input features are independent. Based on the given dataset, it can use any of the 3 models- Gaussian, Multinomial, Bernoulli.
5. **Voting Classifier (VC)**: A voting classifier builds several different base models, averages their output, and then produces predictions. Voting for each estimator output may be incorporated into the aggregating criteria. Hard voting and soft voting are the two categories of voting criteria. In the case of hard voting, the anticipated output class serves as the basis for the vote, but in the case of soft voting, the projected output class probability serves as the basis for the vote.
6. **K-Nearest Neighbours (KNN):** A data point is primarily categorized by the class/label of its neighbors. The parameter "k" in the KNN designates how many nearest neighbors will be considered for most voting. With the use of distance vectors, it determines who its closest neighbors are.
7. **Passive Aggressive (PA)**: The passive-aggressive classifier belongs to the class of online learning algorithms. It can manage vast data sets and alter its model across any new instance. The algorithm alters its weights when any fresh information is received. The regularization parameter, C, solves the trade-off between the margin's size and the number of misclassifications. The classifier scrutinizes a new instance at each iteration and modifies its weight after determining whether it was appropriately classified. There is no change in weight if it is correctly classified.
8. **Multilayer Perceptron (MLP):** An ANN (Artificial Neural Network) called a multilayer perceptron (MLP) is made up of many interconnected layers of perceptron-like neurons. It comprises an input layer, an output, and multiple hidden layers. Each neuron in an MLP processes incoming data and uses weighted connections to send the output to the following layer. To reduce prediction errors, training an MLP entails modifying the weights of these connections using methods like gradient descent and backpropagation.

Performance Matrices: Several methods are available to evaluate the performance of a machine learning model. Some of them used in this paper are explained below:

1. **Confusion matrix -** With two or more output classes, a confusion matrix provides a way to gauge how effectively a machine learning classification algorithm performs. Actual values and predicted values can have below four combinations.

TP-True Positive: The predicted output is positive, and the actual output is also positive.

TN-True Negative: The predicted output is negative, and the actual output is also negative.

FP-False Positive: The predicted output is positive, but the actual output is negative.

FN-False Negative: The predicted output is negative, but the actual output is positive.

	Predicted :1(Positive)	Predicted: 0(Negative)
Actual: 1(Positive)	**TP**	**FN**
Actual: 0(Negative)	**FP**	**TN**

Fig 2. Performance Confusion matrix.

2. **Matthews Correlation Coefficient (MCC) -** The actual class and the predicted class can be considered as two (binary) variables, and their correlation coefficient can be calculated, similar to computing the correlation coefficient between any two variables. This is an alternate and perhaps more elegant method for binary classification when both classes are important. The correlation between the true-to-predicted value changes the prediction accuracy. The Matthews Correlation Coefficient (MCC) is the name given to the phi-coefficient when it is used with classifiers.

If the classifier is flawless (FP and FN are 0), the MCC score 1 denotes a fully positive association. However, if the classifier continually misclassifies (TP and TN are 0), we get a value of -1, which is a perfect negative correlation. The range of MCC is from -1 to 1. No class is more important than any other since MCC is also completely symmetric; if the positive and negative are flipped, the outcome will remain the same. A high value of MCC (near 1) implies that MCC correctly predicts both classes.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3. **Cohen's Kappa Score (CKS):** This measurement has its roots in psychology.
   When ranking patients as subjects, it measures the degree of agreement between two human raters (such as psychologists). We are more certain that the ratings are accurate when there is a significant degree of agreement between the assessors. Low levels of agreement indicate that the evaluations are unreliable. The kappa score, also called inter-rater dependability, determines the level of agreement between the two raters.

4. **Precision (macro and weighted):** Precision can be interpreted as – how many are positive among all the positive predicted classes.
   $$Precision = \frac{TP}{TP+FP}$$
5. **F-1 Score:** It is challenging to compare two different models with low recall but high precision or vice versa. Therefore, to compare them, F-Score is required. The F-score is a way to calculate recall and precision simultaneously. It supersedes the arithmetic mean by harmonic mean.

$$\boldsymbol{F - measure = \frac{2*Recall*Precision}{Recall+Precision}}$$

6. Recall can be interpreted as - How many are correctly predicted as positive from all the positive classes.

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

Using macro averaging, the multi-class predictions are divided into a number of sets of binary predictions. The related metric is then calculated for each binary case, and the results are averaged.

7. **Receiver Operating Characteristics (ROC):** The ROC curve illustrates how sensitive the classifier model is by displaying the proportion of true positives to false positives. A perfect classifier would have no false positives and a true positive rate of 100%. The precise trade-off between the genuine positive rate and the false-positive rate for a model utilizing various measures of probability thresholds can be determined using ROC curves.

Remember that when the data is balanced, we can use the Macro version, but Micro and weighted versions are preferable when data is imbalanced.

4. Implementation

4.1 About the dataset

In our work, we have used the KDDCup99 dataset, which is widely used and accepted by the research community for detecting security threats in Wireless Sensor networks. The dataset is divided into two parts for training and testing purposes. The training part has 125973 samples, while the testing has 22544 samples. The distribution of attacks in the dataset is as follows:

Table 1. Training and Testing Dataset.

	Training data sample	Testing data sample
Attack (1)	58630	12833
Normal (0)	67343	9711

Table 2. Type of Attack in Dataset.

Type of Attack	Training data Samples	Testing data Samples
Normal	67343	9711
Probe	11656	2885
DOS	45927	9460
U2R	52	67
R2L	995	2421
Total	125973	22544

5. Results

5.1 The implementation for binary classification:

Firstly, binary classification has been done in which normal (non-attack) samples are considered one category and all the attacks collectively are considered in the second category. In binary class classification, class 0 refers to normal and class 1 refers to the attacks. Then, some machine learning algorithms are applied to identify the WSN attacks, including Decision Tree, Logistic Regression, Random Forest, MLP, SVM, KNN, NB and Voting Classifier. The performance of the algorithms mentioned above is measured using MCC, ROC, weighted-average and micro-average values of precision, recall and F-1 score. In this part, we have focused more on MCC for measuring the performance of algorithms as it is symmetric. The comparison shows that MLP performed better than other algorithms, followed by SVM. Figures 3 to 7 show the comparison graphs of various algorithms with different performance measures.

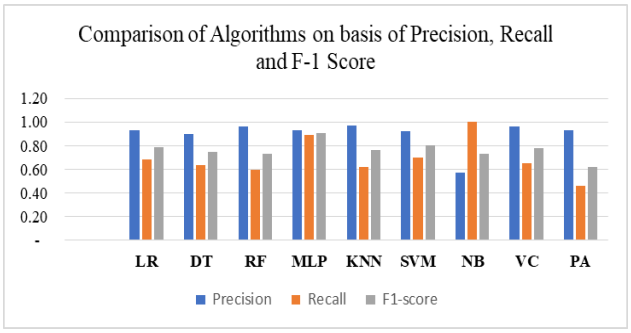

Fig 3. Algorithm Comparison based on precision, recall and F-1 score

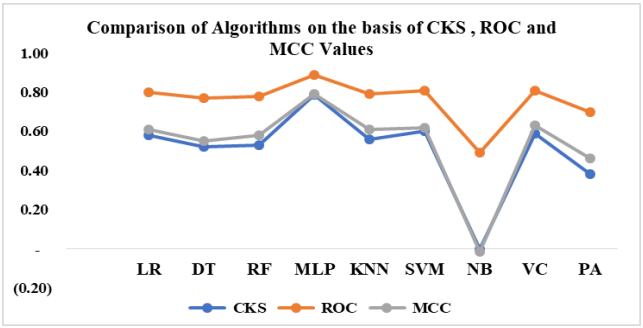

Fig 4. Algorithm Comparison on the basis of CKS, ROC, and MCC values

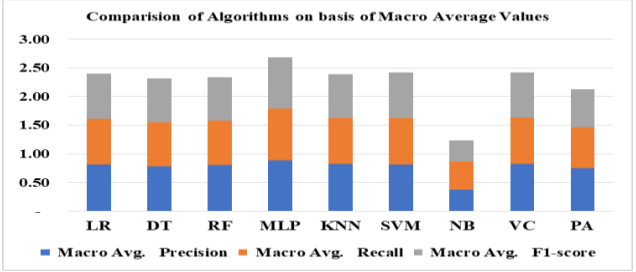

Fig 5. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

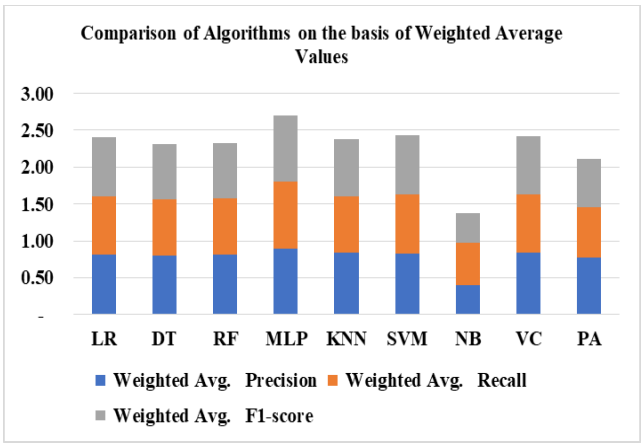

Fig 6. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

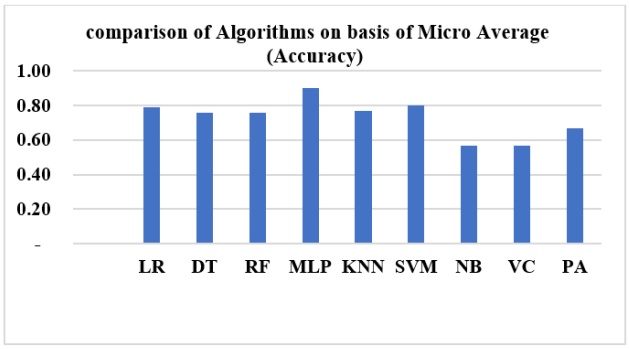

Fig 7. Algorithm Comparison based on accuracy

Figures 3 to 7 show a comparative analysis of various classifiers for TCP binary classification, differentiating normal behavior (0) and all attacks (1) within Wireless Sensor Network datasets, with metrics presented in percentages. It compares eight classifiers: Logistic Regression, Decision Tree, Random Forest, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and a Voting Classifier. The table quantifies each classifier's performance using Precision, Recall, F1-score, Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), and Composite Score (CS), along with their Macro and Weighted Averages. These percentages reflect each model's accuracy, sensitivity, and overall effectiveness in correctly identifying normal activities and various types of network attacks, highlighting their strengths and areas for improvement in network security.

5.2 The implementation of multi-class classification

In the second part, multi-class classification has been done. Each type of attack has been considered as a category. Normal (No attack) samples are class 0, DOS attacks are class 1, Probe attacks are class 2, R2L attacks are class 3, and U2R attacks are class 4. As mentioned above, ML algorithms have been used to categorize and detect these attacks. The performance of the algorithms mentioned above is measured in terms of precision, recall, MCC, ROC

and F-1 Score. LR, SVM and KNN performed better in this multi-class classification than the other algorithms. Fig 8, Fig 9 and Fig 10 show the comparison of algorithms based on different performance measures.

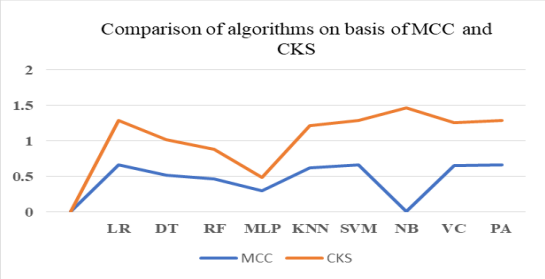

Fig 8. Algorithm comparison based on MCC and CKS

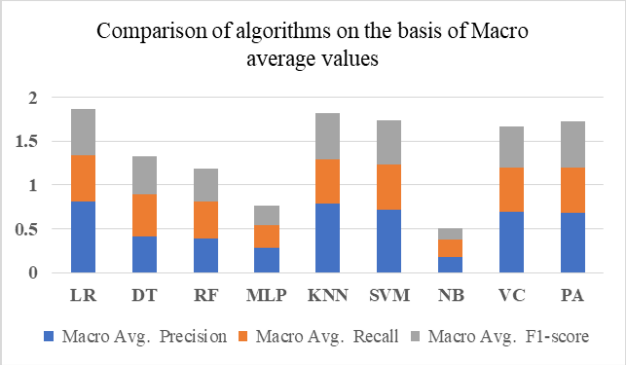

Fig 9. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

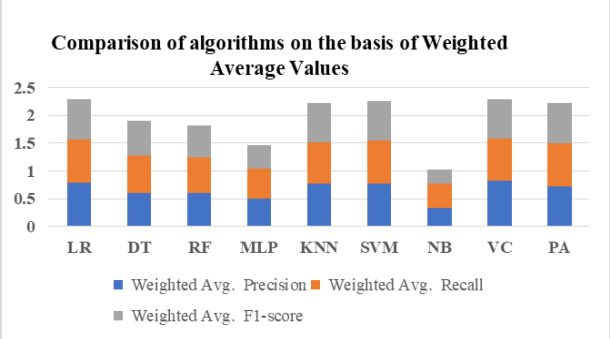

Fig 10. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

Figures 8 to 10 provide a detailed assessment of various classifiers used for multi-class classification in the KDD dataset, where the normal class is treated as an attack type, among others. It evaluates eight classifiers: Logistic Regression, Decision Tree, Random Forest, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and a Voting Classifier. The table presents key performance metrics on a macro and weighted scale, including Matthew's Correlation Coefficient (MCC), Cohen's Kappa Score (CKS), and averages for Precision, Recall, and F1-score. These metrics offer insights into each classifier's ability to correctly identify and differentiate between various types of network behavior, including normal operations and various attack scenarios. The detailed percentages reflect the precision with which each classifier can make predictions, the recall indicates their sensitivity to detecting all relevant instances, and the F1-score provides a balance between precision and recall, all of which are crucial for effective network security in diverse and dynamic attack landscapes.

5.3 The implementation divided the data into four smaller datasets:

In the third part, we have divided the data into 4 smaller datasets, each representing one type of fault. The first data frame is only for DOS attacks, the second for Probe attacks, and the third and fourth for U2R and R2L, respectively. Then, different binary classifiers are applied to classify attacks from normal instances. The results for different faults are given below:

5.3.1 In case of DOS faults, Logistic Regression and MLP perform better than other algorithms followed by SVM and Voting classifier. Figures 11, 12, and 13 are based on different performance measures. The detailed data from the KDD-4 dataset focuses on the performance metrics of various classifiers in detecting DoS attacks, presenting a nuanced view of each model's effectiveness. The classifiers evaluated include Logistic Regression, Decision Tree, Random Forest, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and a Voting Classifier. Metrics such as Precision, Recall, F1-score, Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), and Composite Score (CS) are thoroughly analyzed. The Macro and Weighted Averages for Precision, Recall, and F1-score are also provided, offering a comprehensive look at the classifiers' performance. For instance, MLP shows notable effectiveness with high scores across most metrics, indicating its robust capability in detecting DoS attacks. In contrast, Naive Bayes significantly lags, especially in recall and F1-score, suggesting limited utility in this context. The data collectively highlights the strengths and weaknesses of each classifier, guiding the selection and optimization of models for ensuring robust network security against DoS attacks.

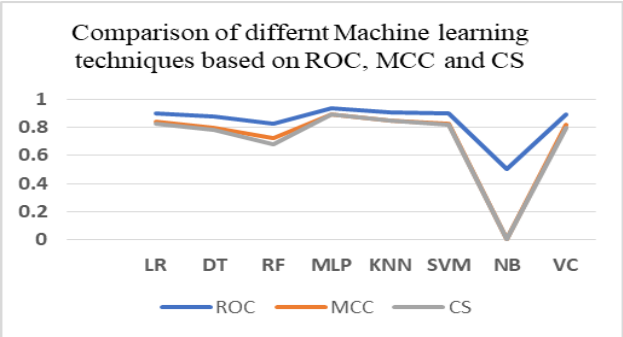

Fig 11. Algorithm comparison based on ROC, MCC and CS

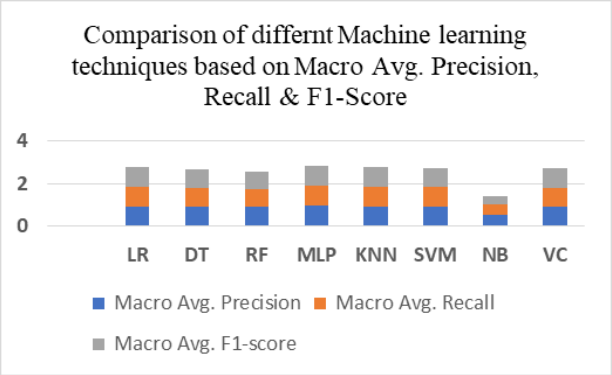

Fig 12. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

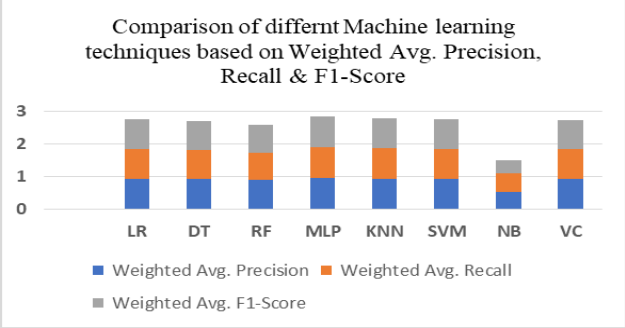

Fig 13. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

5.3.2 In case of Probe faults: Logistic Regression performed better than others. Figures 15, 16, and 17 are based on different performance measures.

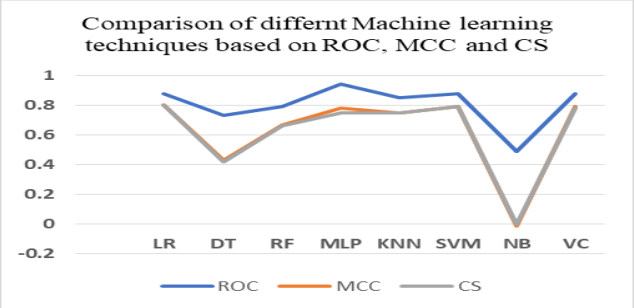

Fig 14. Algorithm Comparison based on ROC, MCC and CS

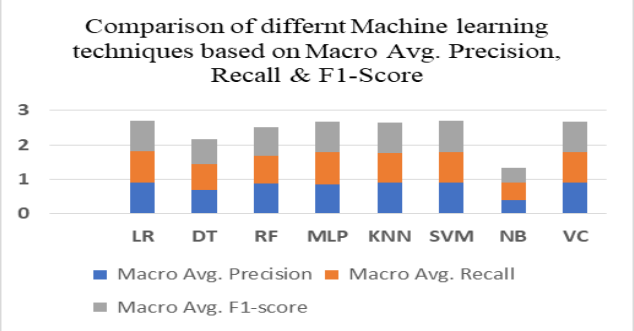

Fig 15. algorithm Comparison based on Macro Average Precision, Recall and F-1 score.

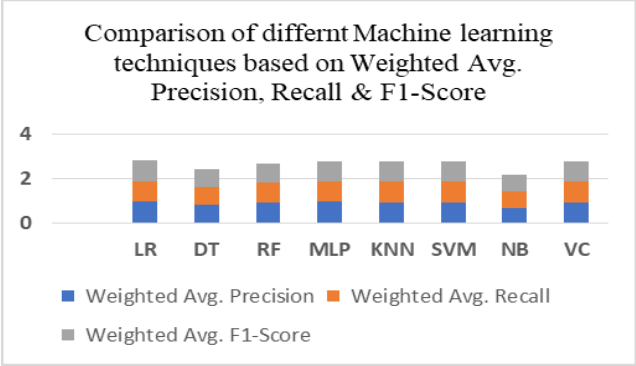

Fig 16. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

Fig 14, Fig 15 and Fig 16 show a comprehensive evaluation of various classifiers' effectiveness in identifying Probe attacks within a network, as indicated through precision, recall, F1-score, Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), and Composite Score (CS), along with both macro and weighted averages. It compares the performance of Logistic Regression, Decision Tree, Random Forest, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and a Voting Classifier. Notably, the MLP stands out for its perfect recall and high F1 score, indicating its strong capability to identify all instances of probe attacks. At the same time, Naive Bayes shows no effectiveness in detecting these attacks. The table also reveals a general trend where most classifiers maintain a good balance between precision and recall, as reflected in their F1-scores and ROC values, except for the Decision Tree and Naive Bayes, which exhibit lower performance metrics. Such detailed metrics are crucial for understanding each classifier's strengths and weaknesses and guiding the development of more robust systems for network security against probe attacks.

5.3.3 Regarding R2L faults, the Decision Tree is performing better. Figures 17, 18, and 19 are based on different performance measures.

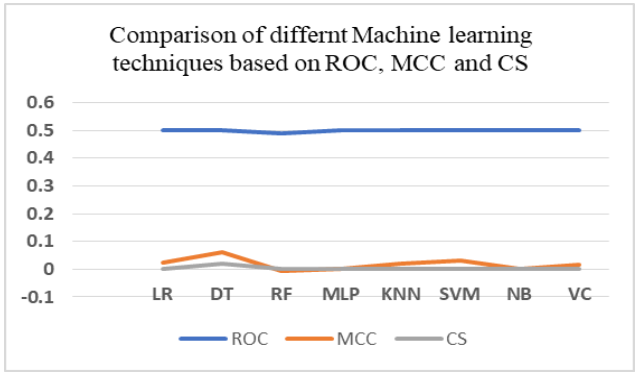

Fig 17. Algorithm Comparison based on ROC, MCC and CS

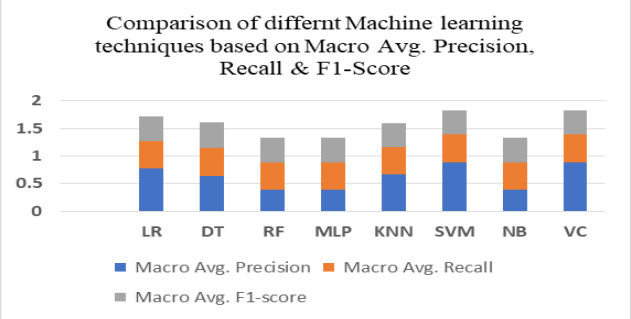

Fig 18. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

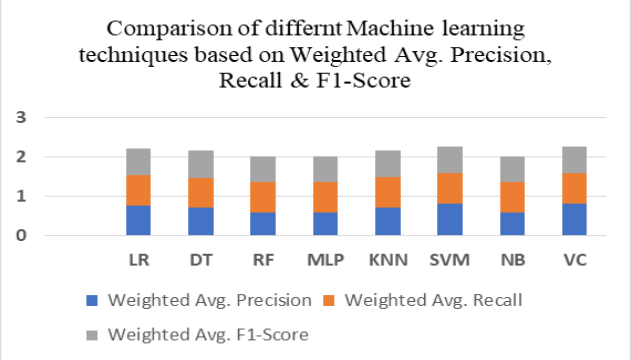

Fig 19. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

Figs. 17, 18, and 19 show the performance of various classifiers in detecting R2L (remote to local) attacks, a specific category of network security breaches. The classifiers assessed include Logistic Regression, Decision Tree, Random Forest, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and a Voting Classifier. The performance metrics provided are Precision, Recall, F1-score, Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), and Composite Score (CS), along with their Macro and Weighted Averages. This table highlights a significant challenge in detecting R2L attacks, as evidenced by the generally low recall and F1 scores across most classifiers, indicating a struggle to correctly identify positive instances of these attacks. While SVM and the Voting Classifier show perfect precision, their recall is 0, reflecting that they failed to identify any actual attacks. Such insights underscore the complexity of effectively detecting R2L attacks and the need for further optimization and possibly more sophisticated or tailored approaches to improve detection rates and ensure robust network security against these intrusions.

5.3.4 In the case of U2R faults, KNN is better than others. Figures 20, 21, and 22 are based on different performance measures.

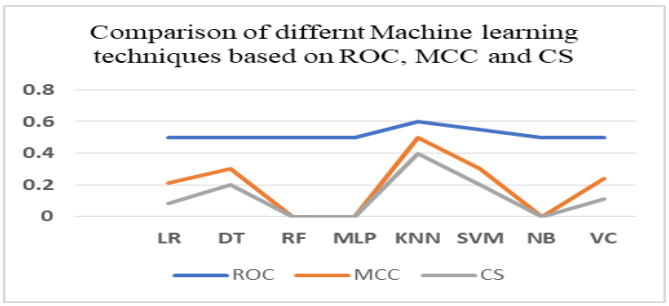

Fig 20. Algorithm Comparison based on ROC, MCC and CS

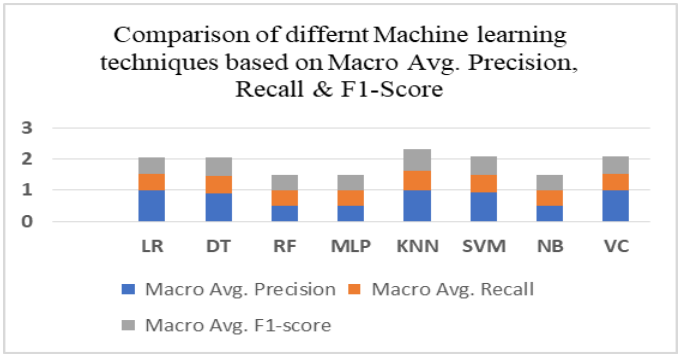

Fig 21. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

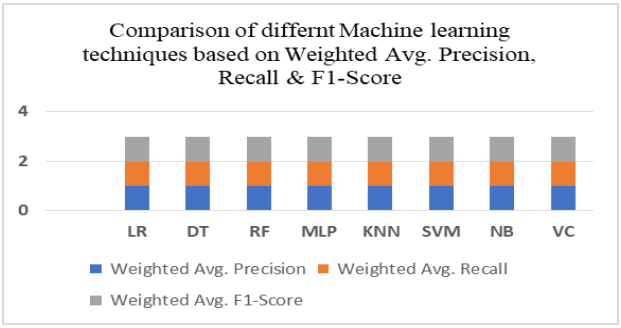

Fig 22. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

Fig 20, Fig 21, and Fig 22 offer an in-depth look at the effectiveness of various classifiers in detecting root (U2R) attacks, a type of security breach where an attacker tries to gain root access from a normal user account. It evaluates classifiers like Logistic Regression, Decision Tree, Random Forest, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and a Voting Classifier. Key performance metrics presented include Precision, Recall, F1-score, Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), and Composite Score (CS), along with their Macro and Weighted Averages. The table reveals a notable challenge in U2R attack detection, with many classifiers struggling to achieve meaningful recall and F1

scores, indicating difficulties in identifying actual attacks effectively. While some classifiers like Logistic Regression and KNN show high precision, their recall rates are low, suggesting they miss many actual U2R attacks. The consistently high weighted averages across precision, recall, and F1-score indicate the models' overall accuracy when accounting for the class imbalance typically present in U2R attack datasets. This detailed analysis underscores the need for more specialized or advanced approaches to improve the detection of these sophisticated and less frequent attack types.

5.4 The implementation using a data balancing technique called SMOTE divided the data into four smaller datasets:

In the fourth part, we applied a data balancing technique called SMOTE to our data. Imbalanced data implies datasets where the target class comprises an unequal distribution of observations, i.e., few classes have a higher number of observations than other classes. So, there is a need to balance the data. For this purpose, we can either increase the number of samples from minority classes (oversampling) or decrease the number of majority classes (undersampling). Synthetic Minority Oversampling Technique (SMOTE), an oversampling technique, picks up a minority class data point randomly and then detects its k number of closest minority classes with the help of Euclidean distance. A line segment is then formed in the feature space by joining that instance and one of its neighbors. At last, a new instance is created between the two points.

After balancing our data using the SMOTE technique, we have compared various ML algorithms to detect the type of attack in WSN. The results for different faults are given below:

5.4.1 In case of DOS faults: MLP performed better than other algorithms followed by KNN, PA and SVM. Figures 23, 24, and 25 are based on different performance measures.

Fig 23, Fig 24 and Fig 25 show performance in detecting Denial of Service (DoS) faults, highlighting the effectiveness of the Synthetic Minority Over-sampling Technique (SMOTE) in addressing class imbalance. It evaluates classifiers including Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Voting Classifier (VC), and Passive Aggressive (PA). The performance metrics presented are Precision, Recall, F1-score (F-1), Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), and Composite Score (CS), along with their Macro and Weighted Averages, and overall accuracy.

Fig 23, Fig 24, and Fig 25 reveal that while most classifiers exhibit high precision, their recall varies, reflecting differences in their ability to identify all actual DoS attacks. MLP demonstrates exceptional performance with high scores across all metrics, particularly in recall and F1-score, indicating its strong capability in accurately identifying DoS attacks. In contrast, Naive Bayes significantly underperforms, particularly in recall and F1-score, suggesting it's less effective in this context. The consistent performance of classifiers like LR, SVM, and VC across most metrics suggests their robustness in detecting DoS attacks when aided by SMOTE. The detailed metrics, including the Macro and Weighted Averages for Precision, Recall, and F1-score, provide a comprehensive view of each classifier's strengths and weaknesses, guiding the selection and optimization of models to ensure robust network security against DoS attacks. Including SMOTE in the analysis underscores its potential to enhance classifier performance in imbalanced data scenarios common in network security.

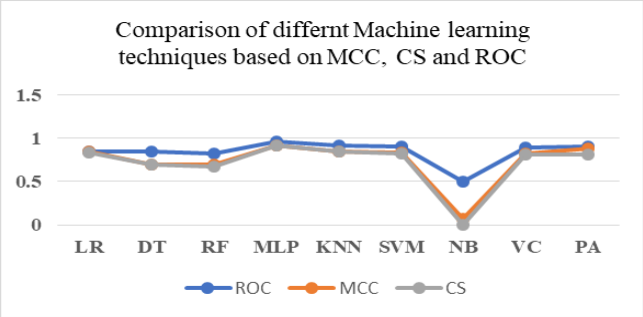

Fig 23. Algorithm Comparison based on MCC, CS and ROC

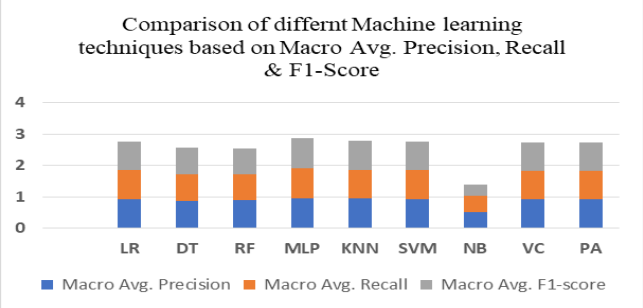

Fig 24. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

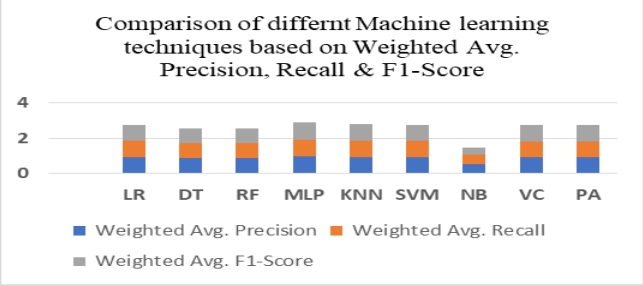

Fig 25. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

5.4.2 In case of Probe faults: Logistic Regression performed better than others, followed by SVM, VC and KNN. Figures 26, 27, and 28 are based on different performance measures.

Fig 26, Fig 27 and Fig 28 show performance in identifying Probe attacks, a type of network intrusion, by utilizing the Synthetic Minority Over-sampling Technique (SMOTE) to handle class imbalance. It examines a range of classifiers: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Voting Classifier (VC), and Passive Aggressive (PA). The metrics used for evaluation include Precision (Pre), Recall, F1-score (F-1), Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), Composite Score (CS), along with Macro and Weighted Averages for Precision, Recall, F1-score, and overall Accuracy.

The detailed performance metrics reveal that classifiers like LR, SVM, and VC excel with high precision, recall, and F1 scores, suggesting a strong ability to detect and classify Probe attacks accurately. Naive Bayes shows zero effectiveness across most metrics, indicating significant limitations. The ROC values and MCC and CS provide additional insights into each classifier's true positive rate and classification quality, respectively. The Macro and Weighted Averages further elaborate on the classifiers' performance across the various classes, reflecting their ability to generalize and perform consistently. The consistent high performance of some classifiers, as indicated by the high accuracy rates, underscores the effectiveness of SMOTE in enhancing model performance for imbalanced datasets. Overall, this detailed analysis helps understand each approach's strengths and weaknesses and guides the optimization and selection of robust models for network security against Probe attacks.

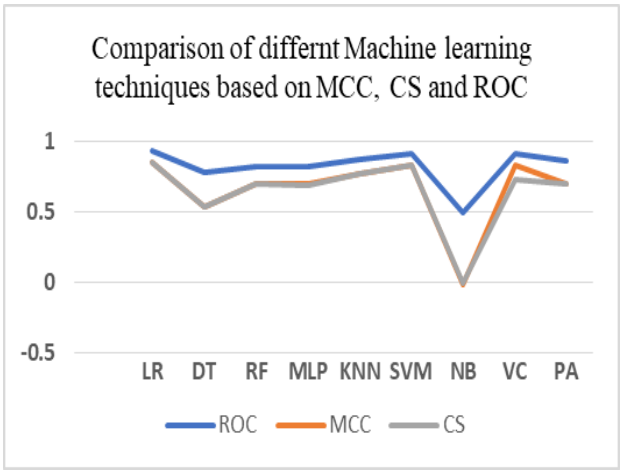

Fig 26. Algorithm Comparison based on MCC, CS and ROC

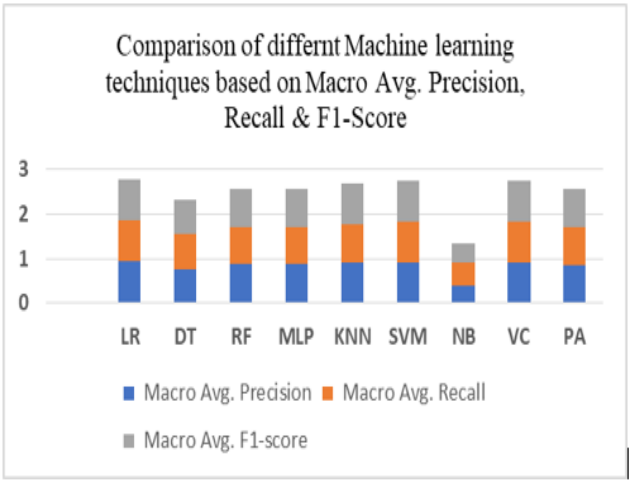

Fig 27. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

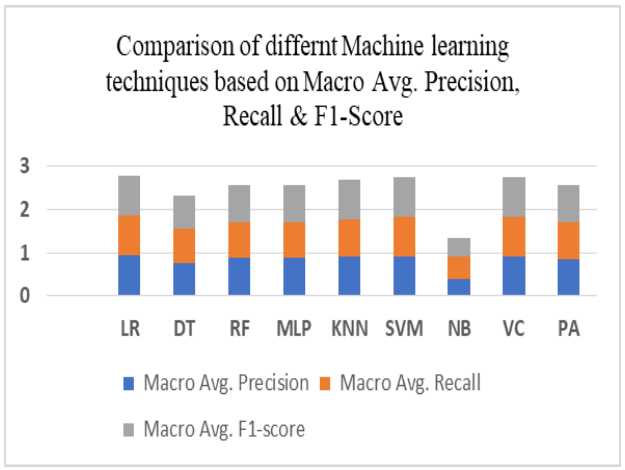

Fig 28. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

5.4.3 In the case of R2L faults, MLP performed better than others, followed by LR and PA. Fig 29, Fig 30, and Fig 31 show this based on different performance measures.

Fig 29, Fig 30 and Fig 31 shows provide a detailed examination of various classifiers' abilities to detect Remote to Local (R2L) attacks using the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalances. The classifiers assessed include Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Voting Classifier (VC), and Passive Aggressive (PA). Performance metrics such as Precision (Pre), Recall, F1-score (F-1), Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), Composite Score (CS), along with Macro and Weighted Averages for Precision, Recall, F1-score, and overall Accuracy are presented.

The data indicates a general struggle among most classifiers to effectively identify R2L attacks, with many showing low recall and F1 scores. For instance, while SVM and MLP

exhibit high precision, their recall rates are notably low, indicating a challenge in detecting actual R2L attacks. The ROC and MCC values provide additional context on the true positive rates and the quality of classification, respectively. The Macro and Weighted Averages further detail each classifier's performance across different classes, reflecting their ability to generalize and perform consistently against various attack types. The consistently low performance across classifiers suggests the complexity and difficulty in detecting R2L attacks, indicating a need for more specialized approaches or advanced techniques to improve detection rates. This detailed analysis is crucial for understanding the limitations and strengths of each classifier and guiding the development of more effective systems for ensuring robust network security against R2L intrusions.

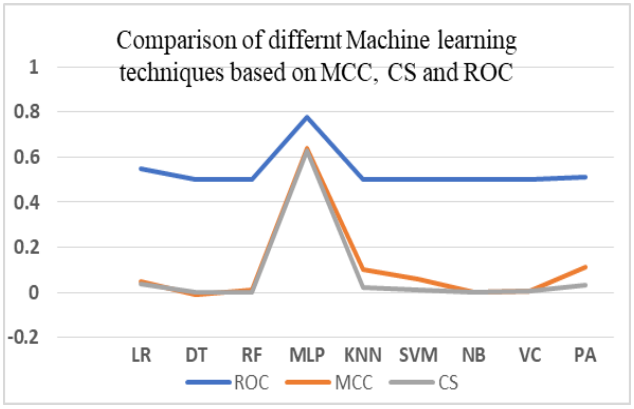

Fig *29*. Algorithm Comparison based on MCC, CS and ROC

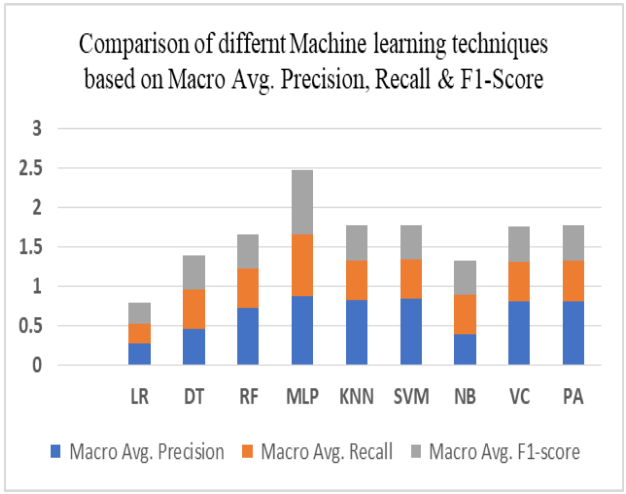

Fig 30. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

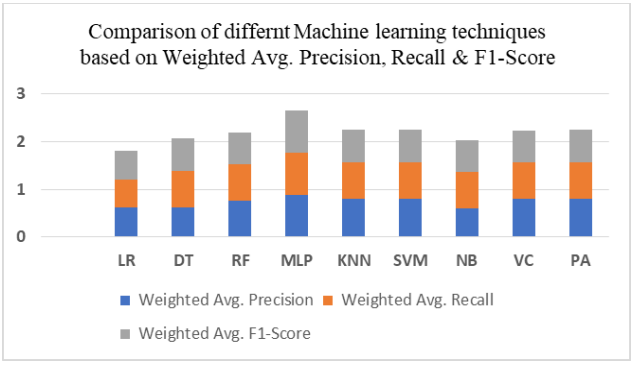

Fig 31. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

5.4.4 Regarding U2R faults, SVM performed better, followed by LR and PA. Fig 32, Fig 33 and Fig 34 show this based on different performance measures.

Fig 32, Fig 33 and Fig 34 shows provide a thorough analysis of the performance of various classifiers using the Synthetic Minority Over-sampling Technique (SMOTE) for detecting Root (U2R) attacks, a challenging and less frequent type of network intrusion. It assesses classifiers like Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Voting Classifier (VC), and Passive Aggressive (PA). The table details metrics such as Precision (Pre), Recall, F1-score (F-1), Receiver Operating Characteristic (ROC), Matthew's Correlation Coefficient (MCC), Composite Score (CS), along with Macro and Weighted Averages for Precision, Recall, F1-score, and overall Accuracy.

This analysis highlights the varied success of different models in detecting U2R attacks. For instance, while LR shows high precision and a relatively good recall, indicating a strong ability to identify U2R attacks accurately, the Decision Tree and Random Forest perform significantly poorly, with zero recall and F1-score. The MLP and KNN display a more balanced performance with moderate recall and F1 scores. With high precision and good recall, the SVM suggests robustness in detecting U2R attacks. In contrast, Naive Bayes demonstrates no effectiveness. The ROC and MCC values provide further insights into the true positive rates and the quality of classification, respectively. The Macro and Weighted Averages elucidate each classifier's performance across different classes, reflecting their ability to generalize and perform consistently. The high accuracy across most classifiers indicates their effectiveness, especially when SMOTE addresses class imbalance. This detailed evaluation is crucial for understanding each classifier's strengths and weaknesses, thereby guiding the development of more effective models for network security against U2R threats.

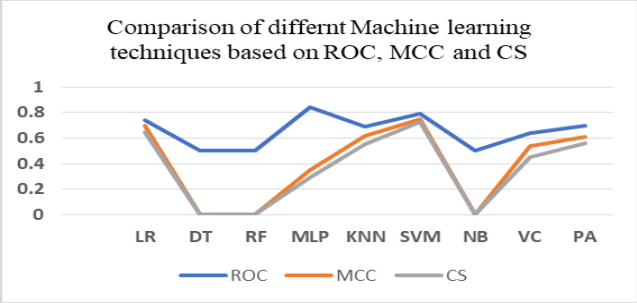

Fig 32. Algorithm Comparison based on ROC, MCC and CS

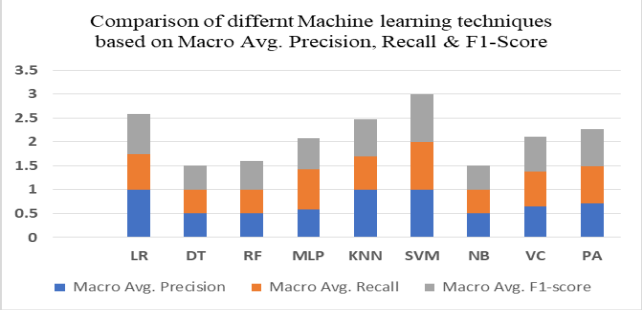

Fig 33. Algorithm Comparison based on Macro Average of F-1 score, Precision-Recall

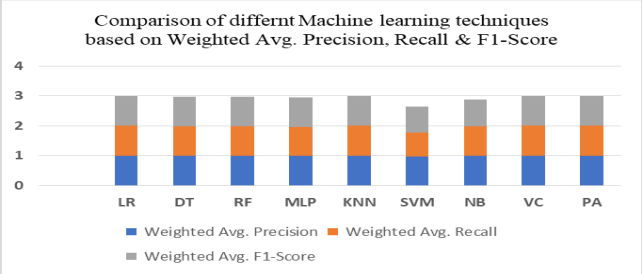

Fig 34. Algorithm Comparison based on Weighted Average of F-1 score, Precision-Recall

6. Conclusion

Fault tolerance in wireless sensor networks (WSNs) is a crucial aspect of ensuring the reliability, availability, and performance of the network despite various types of failures or issues that may occur in its components, such as sensors, communication links, and base stations. WSNs are commonly used in various applications like environmental monitoring, industrial automation, healthcare, etc. Due to their distributed and resource-constrained nature, they are prone to different types of faults, including hardware, communication, and software errors. Achieving fault tolerance in WSNs is an important and challenging task due to the resource constraints of sensor nodes and the dynamic and unpredictable nature of wireless communication. In this paper, we have focused on various WSN attacks and applied a few machine-learning algorithms to detect attacks in WSN. Faults taken into consideration are DOS, Probe, U2R and R2L attacks. The implementation part is divided into four parts. First, binary classification is done to detect attacks and normal events. For this, we have focused on performance matrices like ROC and F-1 score. It is concluded that MLP performed better than other algorithms. In the second part, we have performed multi-class classification to detect five classes (4 attacks and one normal). In this part, we have focused more on MCC to evaluate the algorithms. LR, SVM and KNN performed better than the rest of the algorithms. In the third part, we have divided our dataset into four parts, each of an attack. Then, Machine learning algorithms are evaluated for each of the attacks. It was found that LR and MLP performed better for DOS attacks, LR performed better for Probe, DT for R2L, and KNN performed better for U2R. In the fourth part, a data balancing technique is applied on four subsets of data and attacks are detected using a different machine learning algorithm. It was found that MLP performed better than other algorithms for DOS and R2L, LR performed better for Probe, and SVM performed better for U2R.

In future work, we can apply other data balancing techniques to balance the data and find out which works best for these attacks in WSN. Further, hyperparameter tuning can be done to find out the set of parameters that works best for WSN attacks.

Author contributions

Neha Jagwani: Conceptualization and design of work, Data analysis and interpretation, Writing-Original draft preparation.

Dr Poornima G: Conceptualization of work, Critical revision of the article.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Elsadig, Muawia A. "Detection of Denial-of-Service Attack in Wireless Sensor Networks: A lightweight Machine Learning Approach." IEEE Access (2023).

[2] Lai, Trinh Thuc, Tuan Phong Tran, Jaehyuk Cho, and Myungsik Yoo. "DoS attack detection using online learning techniques in wireless sensor networks." Alexandria Engineering Journal 85 (2023): 307-319.

[3] Ahmad, Rami, Raniyah Wazirali, and Tarik Abu-Ain. "Machine learning for wireless sensor networks security: An overview of challenges and issues." Sensors 22, no. 13 (2022): 4730.

[4] Salmi, Salim, and Lahcen Oughdir. "Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network." Journal of Big Data 10, no. 1 (2023): 1-25.

[5] Dener, Murat, Celil Okur, Samed Al, and Abdullah Orman. "WSN-BFSF: A New Dataset for Attacks Detection in Wireless Sensor Networks." IEEE Internet of Things Journal (2023).

[6] Gebremariam, Gebrekiros Gebreyesus, J. Panda, and S. Indu. "Secure localization techniques in wireless sensor networks against routing attacks based on hybrid machine learning models." Alexandria Engineering Journal 82 (2023): 82-100.

[7] Ismail, Shereen, Zakaria El Mrabet, and Hassan Reza. "An Ensemble-Based Machine Learning Approach for Cyber-Attacks Detection in Wireless Sensor Networks." Applied Sciences 13, no. 1 (2022): 30.

[8] Abidoye, Ademola P., Ibidun C. Obagbuwa, and Nureni A. Azeez. "Mitigating denial of service attacks in fog-based wireless sensor networks using machine learning techniques." Journal of Data, Information and Management 5, no. 4 (2023): 207-225.

[9] Ismail, Shereen, Diana W. Dawoud, and Hassan Reza. "Securing Wireless Sensor Networks Using Machine Learning and Blockchain: A Review." Future Internet 15, no. 6 (2023): 200.

[10] Mounica, Mandala, R. Vijayasaraswathi, and R. Vasavi. "RETRACTED: Detecting Sybil Attack In Wireless Sensor Networks Using Machine Learning Algorithms." In IOP Conference Series: Materials Science and Engineering, vol. 1042, no. 1, p. 012029. IOP Publishing, 2021.

[11] Gebremariam, Gebrekiros Gebreyesus, J. Panda, and S. Indu. "Localization and detection of multiple attacks in wireless sensor networks using artificial neural network." Wireless Communications and Mobile Computing 2023 (2023).

[12] Kousar, Samina, Humaira Ashraf, and N. Z. Jhanjhi. "Detection of Sinkhole Attack in Wireless Sensor Networks Using Machine Learning."

[13] Ifzarne, Samir, Hiba Tabbaa, Imad Hafidi, and Nidal Lamghari. "Anomaly detection using machine learning techniques in wireless sensor networks." In Journal of Physics: Conference Series, vol. 1743, no. 1, p. 012021. IOP Publishing, 2021.

[14] Saleh, Hadeel M., Hend Marouane, and Ahmed Fakhfakh. "Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning." IEEE Access (2024).

[15] Yu, Dongxian, Jiatao Kang, and Junlei Dong. "Service attack improvement in wireless sensor network based on machine learning." Microprocessors and Microsystems 80 (2021): 103637.

[16] Annapurna, V. Gowtami, K. Anusha, CH Kamala Varsha, M. Deepthi, and G. Keerthi. "Detection of Network Layer Attacks in Wireless Sensor Network." Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146 9, no. 1 (2023): 42-48.

[17] Gebremariam, Gebrekiros Gebreyesus, J. Panda, and S. Indu. "Blockchain-Based Secure Localization against Malicious Nodes in IoT-Based Wireless Sensor Networks Using Federated Learning." Wireless Communications and Mobile Computing 2023 (2023).

[18] Bagwari, Ashish, J. Logeshwaran, K. Usha, R. Kannadasan, Mohammed H. Alsharif, Peerapong Uthansakul, and Monthippa Uthansakul. "An Enhanced Energy Optimization Model for Industrial Wireless Sensor Networks Using Machine Learning." IEEE Access (2023).

[19] Oleiwi, Haider W., Doaa N. Mhawi, and Hamed Al-Raweshidy. "A meta-model to predict and detect malicious activities in 6G-structured wireless communication networks." Electronics 12, no. 3 (2023): 643.

[20] Abhale, Ashwini B. "Deep Learning Perspectives to Detecting Intrusions in Wireless Sensor Networks." International Journal of Intelligent Systems and Applications in Engineering 11, no. 2s (2023): 18-26.

[21] Alqahtani, Mnahi, Abdu Gumaei, Hassan Mathkour, and Mohamed Maher Ben Ismail. "A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks." Sensors 19, no. 20 (2019): 4383.