IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

Design and Development of a Detection System for DoS and DDoS Attacks on WSNs Using Machine Learning

Harshali Patil[1], Gayatri Mestry[2], Umang Maurya[3], Nikita Mali[4]

Head of Department, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India[1]

Student, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India[2]

Student, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India [3]

Student, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India [4]

Abstract: This paper introduces a Machine Learning-based detection system for Denial of Service attacks on WSNs providing robust cybersecurity to these vulnerable systems. The class imbalance problem is quite significant in the WSN-DS dataset, so SMOTE will be used to create synthetic samples to balance the distribution of instances for attack and normal data. Then, feature selection is used which guides the search for relevant attributes to effectively detect attacks. Further, three different machine learning models were trained and evaluated: Logistic Regression, Decision Tree, and REPTree, measuring them in terms of accuracy, precision, recall, and F1-score. This study illustrates that this approach works towards correctly identifying the diverse categories of DoS attacks very efficiently and creates grounds for more effective security strategies for WSNs.

Keywords: Wireless Sensor Networks, Denial of Service, Distributed Denial of Service, Machine Learning.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of spatially distributed sensors that monitor environmental conditions and transmit the collected data to a central location for processing. These sensors work together to ensure stable and efficient communication across a network spread over various locations [1]. Due to their collaborative nature, WSNs are ideal for measuring environmental parameters such as humidity, temperature, sound, and pollution levels. The sensor nodes in WSNs are designed to be low-power, compact, and cost-effective, with capabilities for sensing, wireless communication, and computation [2]. The most common applications of WSNs include area monitoring, healthcare monitoring, habitat monitoring, forest fire sensing, landslide detection, water quality monitoring, etc. All this said, WSNs are also particularly susceptible to cyber attacks due to their deployment in hostile environments and their wireless nature which is why research about the possible cyber attacks, their detection, prevention and studying about the countermeasures against them is necessary [3].

In this paper, there will be a focus on Denial of Service and Distributed Denial of Service attacks against WSNs and how to detect them using machine learning so as to avoid them with as much efficiency as possible. Machine learning works by using historical data to learn patterns which enable the Machine learning model to then be able to identify an attack or anomalies which could potentially be a DoS or DDoS attack. Various machine learning techniques have shown to work well in this regard [4].

The aim of this study is to look into the existing methods of detection and to design a new system which will be able to detect DoS and DDoS attacks while also trying to improve upon the gaps in the existing systems. The proposed system is made up of a balanced WSN-DS dataset and supervised learning technique REPTree. It is also primarily assessed with its F1 score.

II. BACKGROUND

The main components of each sensor node in WSNs are– Sensing unit, Processing unit, Transceiver, Power Unit. The sensing unit is used to sense the physical quantities as required and transformed into a digital format using an Analog to Digital Converter, after which the digital data undergoes any processing required in the Processing unit.

DOI: 10.17148/IJARCCE.2025.14444

The Transceiver is used to transmit and receive this data between other sensor nodes and the Power unit is used to supply the node with the required power [2].

WSNs face several hardware limitations due to their resource-constrained nature. Sensor nodes are typically equipped with minimal processing power, memory, and battery capacity. These limitations mean that nodes can only perform basic computations and store limited amounts of data. The small battery life is especially a concern, as it restricts the operational lifespan of the network, requiring energy-efficient communication and data-handling strategies. Additionally, the limited processing and memory capacity can reduce the ability of nodes to handle large amounts of data or perform more complex tasks, making careful resource management essential [5].

WSNs are usually organized in a cluster based hierarchical topology where it groups its nodes into sensor nodes, cluster heads (CH) and base stations (BS). In this system, the sensor nodes are responsible for collecting the required information from their surroundings, after which they transmit it to their respective cluster heads. These cluster heads aggregate the received data to reduce its size while retaining its informational value and then transmit it to the base station or sink. This helps with minimizing the amount of data that gets transmitted as well as the number of nodes involved in the transmission and helps to maintain energy consumption. Due to the nodes of WSNs being resource constrained in terms of battery, memory and processing, they use the Low Energy Adaptive Clustering Hierarchy (LEACH) routing protocol [6].

In each communication round, the cluster heads are selected randomly, this random rotation helps avoid the overburdening of any single node, extending the network's lifetime. LEACH functions in two primary phases: the setup phase, where clusters are formed and cluster heads are chosen, and the steady-state phase, where data transmission and aggregation occur.

Despite its advantages, LEACH has notable limitations, particularly in larger WSN deployments. The random cluster head selection can lead to an imbalance in energy usage or result in low-energy nodes being chosen, diminishing network performance. Moreover, LEACH assumes that all nodes have sufficient transmission power to communicate directly with the base station, an assumption that may not hold in extensive or resource-constrained networks, limiting its scalability and applicability [7].

These hardware limitations and the LEACH protocol increase WSNs' vulnerability to intruders as it makes it easy for them to insert an intruder node. After which, they use that node to introduce malicious messages, bombard the network with unnecessary or illegitimate information to weaken it or drop data packets [6].

Lastly, DoS and DDoS attacks work by disrupting the normal functioning of the network by overwhelming the sensor nodes or communication channels with excessive traffic. This can lead to the depletion of energy resources, data loss, and overall network failure, severely impacting the effectiveness of WSNs in their respective applications. The inherent resource constraints of WSNs, such as limited battery life and computational power, make them particularly vulnerable to such attacks [4].

III. LITERATURE REVIEW

Most of the threats and attacks against wireless networks are similar to wired networks, and only enhanced due to their wireless nature. Wireless networks are more vulnerable to threats and attacks because unguided transmission mediums are more susceptible to security compromises than guided transmission mediums. The fact that wireless networks use broadcasting as a way of communication makes them more vulnerable to eavesdropping. Another challenge in employing an efficient security system for WSNs is caused due to the size of the sensor nodes and consequently the processing power, memory and type of tasks that are expected of them [3].

In the early stages of research about WSNs, Wood, A. D., & Stankovic, J. A. (2002) provided a comprehensive survey of the different DoS attack types specific to WSNs and discussed various countermeasures against them. There are several types of DoS attacks that can be performed at different layers of WSNs. At the physical layer the attacker could perform jamming and tampering, at the link layer, collision, exhaustion, unfairness, at the network layer, neglect and greed homing, misdirection, black holes and at the transport layer this attack could be performed by malicious flooding and desynchronization.

They also discussed how varied the applications of WSNs are such as military scenarios wherein they can be used for tracking enemy troop movements, monitoring a secured zone or battlefield conditions, since the sensors are small, they can be transported easily with lower risk with an airplane. Other applications include impromptu communication

DOI: 10.17148/IJARCCE.2025.14444

networks for rescue personnel at disaster sites, locating casualties, monitoring conditions at the rims of volcanoes or along earthquake faults or around critical water reservoirs. They can also be used to detect chemical or biological threats in crowded areas like airports or stadiums. These kinds of applications make the security of WSNs even more important [6].

"Detection of DoS Attack in WSNs: A Lightweight Machine Learning Approach," proposes a simple Machine Learning algorithm with which to detect DoS attacks in WSNs, which is a decision tree method applied on an the WSN-DS dataset which the author enhances with the help of Gini feature selection to reduce the dimensionality to boost performance. This paper also focuses on the minimization of the computational overhead that often comes with complex detection methods. The proposed method performs well with a high accuracy of 99.5% with a significantly lower computational overhead as compared to other similar, lightweight classifiers like Random Forest, XGBoost and K nearest neighbors. It was also observed that all the classifiers being compared performed better when working on the enhanced version of the dataset, which tells us that feature selection plays an important role and must not be neglected. Owing to this result, the author also suggested the use of the enhanced dataset over the original WSN-DS dataset.

The reason why this paper focuses on minimizing computational overhead is due to the fact that the sensor nodes that make up WSNs are very simple devices that cannot handle heavy processing and do not have a long battery life.

The limitation of this study, as mentioned by it, is that the proposed architecture was only trained on a single dataset. Future work can involve the use of different datasets. Another observation of this study is that WSN-DS is an unbalanced dataset and its balancing is a gap that can be worked on to improve performance [8].

DoS attacks can cause excessive overhead which can lead to resource exhaustion or delay of messages due to congestion in the sensors and by extension, the entire network. Historically, the major approaches that have been used for IDS have been signature, anomaly or hybrid based and Machine learning techniques offer an alternative to these approaches. Quincozes et al. compared various such machine learning techniques in their research including supervised and unsupervised learning both; they compared the performance of these techniques in their ability to detect DoS attacks in sensor networks. They used metrics such as accuracy, recall, precision, F1-Score, and processing time. Along with this, they also performed a comparison of the different values of K to find the best one for each attack and unsupervised detection technique.

They found that supervised learning yielded better results with the REPTree algorithm presenting an F1-Score of 95.69% for detecting blackhole attacks and also a faster computational speed of 0.931µs on average to classify a sample.

Quincozes et al. focussed on DoS attacks in their study, specifically on Flooding, Grayhole and Blackhole attacks. All of these attacks work by sending a large amount of Advertising Cluster Head (ADV CH) messages to the sensor nodes with high transmission power to ensure that the messages reach as many sensor nodes as possible. This causes the sensor nodes to spend a lot of power and tricks them into accepting the intruder as their cluster head. After which, the intruder node can also drop data packets.

They also compared three feature selection techniques- namely Information Gain (IG), Gain Ratio (GR), and OneR and found OneR to be the best one for the ML algorithms they used and the attacks on which their comparisons were based. Feature selection allowed them to observe that (Is_CH) and (ADV_CH_Sent) are the most frequent in each combination of features in all the 3 attacks which makes sense since they all depend on becoming the Cluster Head and they all start with sending out many Advertising Cluster Head messages.

Finally, they concluded that one single solution is not the best for all the attacks and that is why they would like to explore combining multiple algorithms into a single IDS [6].

Ahmad et al. were aiming to improve network performance by updating the LEACH protocol and combine feature selection with machine learning to better DoS detection in WSNs while maintaining energy efficiency to extend the network's lifetime. They noted that the increase in the use of WSNs in the future will lead to an increase in the amount of data being transmitted which will cause it to become more complex than it is presently. Hence, there is a need to use machine learning along with feature selection techniques for detecting DoS attacks with the help of only the necessary data. So for feature selection, the authors compared different techniques like Water Cycle (WC), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Harmony Search (HS), and Genetic Algorithm (GA) and chose Water Cycle optimization in order to minimize computational overhead while increasing detection accuracy as evaluation results showed it to display the best performance accuracy. They also compared Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Deep Learning.

DOI: 10.17148/IJARCCE.2025.14444

The research was conducted on simulated attacks (Blackhole, Grayhole, Flooding, and Scheduling) based on the WSN-DS dataset and these attacks are performed on a LEACH based WSN architecture. They used a simulated proposed clustering algorithm, CH_Rotations, which improved the network lifetime by 30% compared to the standard LEACH protocol. The WC algorithm was evaluated for its ability to select the best subset of features with the lowest computational cost while maintaining high detection accuracy, it performed better than PSO, SA, HS, and GA by achieving an average accuracy improvement of 2%, 5%, 3%, and 3% compared to the other algorithms, respectively. Out of the machine learning algorithms, Decision Tree was chosen for its performance. WC with the Decision Tree classifier achieved an accuracy of 100% using only one feature, demonstrating that feature selection reduced computational load significantly without sacrificing detection accuracy.

However, the techniques being compared were tested on the WSN-DS dataset which is not a balanced dataset as we found in [8]. Also, tests for DoS attacks were only limited to Blackhole, Grayhole, Flooding, and Scheduling [9].

Yao et al. introduced a method by integrating PCA for dimensionality reduction along with a Deep Convolutional Neural Network (DCNN) for effective detection of DoS anomalies in WSN data traffic and also addressed both performance and resource constraints that WSN nodes present. They made it lightweight by incorporating depthwise separable convolutions, an attention mechanism and Global Average Pooling (GAP). GAP allows for efficient feature extraction while not overwhelming the WSN nodes. Because of this, the model also performed well when compared against other machine learning and deep learning methods, especially in terms of accuracy and recall. The use of PCA along with a DCNN also helps to reduce the complexity of the model and hence improves its detecting capability. The reason why this is that using techniques such as GAP and separable convolution reduces the size of the model significantly, especially when compared to the other CNN models.

In this study, Yao et al. used PCA to reduce the dimensionality of the data by eliminating redundant features while retaining the ones with a higher impact. After which, the proposed DSNN structure uses separable convolution which is an attention mechanism to replace pooling layers which helps to avoid feature loss and GAP to replace fully connected layers. The model was evaluated using the KDDcup99, NSL-KDD, and UNSW-NB15 datasets and it focused on binary classification, which entailed differentiating between regular traffic and DoS traffic.

Although the method proposed in this paper was a deep learning approach, it helped us to gain valuable insights regarding the performance benchmarks. Contrary to common perception about the computational heaviness of deep learning models, this study presented a lightweight deep learning model. We intend to compare our model's performance with it to see if a machine learning model, which is inherently lighter than deep learning models, can outperform a specifically lightweight deep learning approach. The proposed DCNN is significantly smaller than CNN or SNN models, reduced to 87.46% of a typical CNN model and 84.77% of a typical SNN model. This reduction is due to the decrease in the number of parameters by 69,496 in a CNN model and 55,483 in an SNN model caused by the use of separable convolution and GAP.

However, a limitation of this study is that the datasets used for evaluation (KDDcup99, NSL-KDD, UNSW-NB15) are not specific to WSNs [10].

The proposed STLGBM-DDS system in [11] demonstrates a high accuracy of 99.95% in detecting DoS attacks on WSNs and an F1 score of 99.95%, surpassing other machine learning and deep learning models. It addresses data imbalance in WSN datasets through the SMOTE + Tomek-Links (STL) method, enhancing classification performance for underrepresented classes. The LightGBM algorithm shows superior performance in detecting WSN-specific intrusions compared to traditional methods and other hybrid deep learning approaches. Feature selection using the Information Gain Ratio reduces computational complexity and increases classification accuracy by focusing on the most relevant features.

The study used the WSN-DS dataset, on which it performed preprocessing by normalizing the data and encoding categorical values with One-Hot Encoding, followed by feature selection using Pearson Correlation Coefficient and Information Gain Ratio, reducing the dataset to the 13 most relevant features. The STL technique balanced the minority class without overfitting. The LightGBM algorithm was employed for classification due to its speed and efficiency in handling large datasets.

Future work will expand the evaluation to different datasets for validation. The model could be improved by incorporating hybrid techniques like combining LightGBM with CNN, LSTM, or GRU, and exploring additional oversampling and undersampling methods. Plans include integrating Explainable Artificial Intelligence (XAI) to enhance transparency and reliability [11].

DOI: 10.17148/IJARCCE.2025.14444

The proposed multilayer detection system in [12] effectively detects internal DoS attacks in WSNs using two machine learning models: Naive Bayes for binary classification and LightGBM for multi-class classification. The system incorporates a mobile robot for routing critical information, reducing communication overhead and enhancing attack detection speed. This approach achieves high accuracy and low false alarm rates, making it suitable for resource-constrained WSN environments.

The system uses Naive Bayes for the first layer binary classification at monitor nodes, achieving 97.25% accuracy, LightGBM for second layer multi-class classification at the base station, achieving an accuracy of 99.3% and F1 score of 99%. It uses mobile robot assistance to collect and route critical attack information. Mutual Information is used for feature selection to reduce dataset dimensionality, with the first layer using an unbalanced dataset and the second layer using a balanced dataset. The first-layer NB model had a processing time of 0.006 seconds for 2,000 samples, with an average prediction time of 0.828 µsec per sample. The second-layer LightGBM model had a processing time of 1.532 seconds for 26,000 samples, with an average prediction time of 2.9 µsec per sample.

Future work includes extending detection capabilities to external attacks and other cyber-attacks, exploring more robust machine learning algorithms, and incorporating additional robots [12].
The proposed method in [13] integrates rule-based and ML techniques to efficiently detect DoS attacks in WSNs. Initially, a rule-based method is employed to distinguish normal traffic flow from potential attack flow. If an attack is detected, ML models are subsequently applied to classify the specific type of DoS attack. This approach enhances system efficiency by avoiding the invocation of ML models for normal traffic, thus conserving computational resources.

Among the ML models tested, the Decision Tree classifier performed best, achieving high accuracy across all attack types, while the Support Vector Machine showed lower performance. The integrated approach leverages rule-based logic to identify normal flows, applying ML classifiers only when necessary, which reduces computational costs. This provides a lightweight and efficient solution for real-time detection of DoS attacks, making it suitable for resource-constrained WSN environments.

The Decision Tree classifier significantly outperformed the SVM as the accuracy and F1 scores for the Decision Tree classifier were notably high: 99.5% accuracy and 99.5% F1-score on average for all the 4 attacks.

Future work, as mentioned, is to enhance the model by exploring additional lightweight classifiers to maintain high accuracy while improving performance speed. Extending the model to detect external attacks or other types of network-layer threats beyond DoS could also be beneficial [13].

The paper “An Advanced Feature Selection Approach to Improve Intrusion Detection System using Machine Learning” proposes a hybrid feature selection method to enhance intrusion detection by selecting the most relevant features, thereby improving accuracy and reducing computational complexity. The model combines Entropy-based Infinite Feature Selection (E-IFS) with Eigenvector Centrality to refine the feature set for intrusion detection, leading to fewer false alarms and better performance.

This novel approach, tested on the NSL-KDD and KDD-CUP99 datasets, showed superior performance compared to traditional models. The E-IFS algorithm, combined with Eigenvector Centrality, selects the most informative features, improving classifier efficiency and effectively detecting DoS and other types of attacks with higher accuracy using fewer features. This hybrid feature selection reduces both training time and computational complexity, making it more suitable for real-time IDS.

The model focuses on two key datasets (NSL-KDD and KDD-CUP99), with plans to validate its performance on more diverse and recent datasets for broader applicability. Future research could integrate deep learning with the proposed feature selection technique to handle more complex attack patterns in IoT and large-scale networks as stated in the literature. Additionally, adaptive feature selection methods that dynamically adjust based on network conditions could further enhance real-time performance.

The proposed method includes four key phases: data selection (NSL-KDD and KDD-CUP99), preprocessing (normalization and encoding using min-max normalization), feature selection (E-IFS and Eigenvector Centrality), and classification (using K-Nearest Neighbors, Artificial Neural Network, and Decision Tree classifiers). For both the datasets, DT and KNN achieved the highest accuracies and F1 scores with DT having a higher accuracy (96.53% and 99.67%) and KNN having a higher F1 score (97.29% and 99.81%) [14].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

TABLE I PAPER TITLE, FINDINGS AND GAPS

Paper Title	Publication Year	Findings	Gaps
Wireless sensor networks in the internet of things: review, techniques, challenges, and future directions [1].	2023	Highlights the challenges faced in WSNs, including security issues and the potential for using ML for intrusion detection.	Lacks specific focus on DoS/DDoS attack detection using ML in WSNs/
Detection of DoS Attack in WSNs: A Lightweight Machine Learning Approach [8].	2023	a. The paper states that deep learning approaches aren't adequate solutions for detection of attacks in WSNs due to the amount of overhead computation that is required by them. b. Due to the increasing diversity of network attacks, machine learning techniques alone are not sufficient for detection, they should be implemented in combination with other techniques like, other machine learning algorithms or feature extraction. d. The enhanced dataset, after the application of Gini feature selection, gave better results.	a. Only one dataset was used and a future scope is to compare the classifiers on different datasets as well. b. WSN-DS is an unbalanced dataset and wasn't balanced in this study.
An extended evaluation on machine learning techniques for Denial-of-Service detection in Wireless Sensor Networks [6].	2023	a. Excessive overhead from DoS attacks can cause rapid sensor resource exhaustion or increase message delay due to network congestion. In critical scenarios, these attacks may threaten human lives. b. Machine learning (ML) methods are promising for detecting DoS in dynamic environments, reducing the need for network redesign. c. Supervised techniques, like REPTree, outperform unsupervised methods, with an F1-Score of 95.69% for blackhole attack detection. REPTree is the fastest, classifying samples in 0.931µs on average.	a. The study used only the WSN-DS dataset, which may not represent all types of WSNs or IoT environments. The dataset includes only three types of attacks (Flooding, Grayhole, and Blackhole), limiting the generalization of the findings. b. The paper focuses on DoS attacks, but there is limited discussion on how the methods would generalize to other types of network attacks. c. While parameter tuning was investigated, unsupervised techniques consistently performed worse than supervised methods, which may indicate the need for further refinement of these approaches for WSNs.
Feature-Selection and Mutual-Clustering Approaches to Improve DoS Detection and Maintain WSNs' Lifetime [9].	2021	a. The CH_Rotations algorithm improved the network lifetime by 30% compared to the standard LEACH protocol. b. The Water Cycle (WC) algorithm performed better than PSO, SA, HS, and GA. It achieved an average accuracy improvement of 2%, 5%, 3%, and 3% compared to the other algorithms, respectively. c. The WC with the Decision Tree classifier achieved 100% accuracy	a. The research relied on simulations and did not address real-time application, where issues like latency and computational limits may arise. b. The study focused only on specific DoS attacks. Further research is needed to extend detection methods to other network attack types. c. Although WC outperformed other feature selection methods,

DOI: 10.17148/IJARCCE.2025.14444

		using only one feature, demonstrating that feature selection reduced computational load significantly without sacrificing detection accuracy. d. Combining WC with machine learning algorithms, especially DT, improved the accuracy and reduced the complexity of detecting DoS attacks.	the paper could benefit from exploring more recent optimization algorithms to validate WC's effectiveness further.
Traffic Anomaly Detection in Wireless Sensor Networks Based on Principal Component Analysis and Deep Convolution Neural Network [10].	2022	a. Improved Detection: The combination of Principal Component Analysis (PCA) and Deep Convolution Neural Network (DCNN) improves detection of Denial of Service (DoS) attacks in Wireless Sensor Networks (WSNs) by reducing the model's complexity and enhancing its detection accuracy. b. Lightweight Model: The proposed model is lightweight, making it more suitable for deployment in resource-constrained WSN environments without compromising detection accuracy. The DCNN's performance, particularly its accuracy, precision, recall, and F1-score, outperformed existing models like CNN and SNN, especially on datasets like KDDcup99, NSL-KDD, and UNSW-NB15. c. Reduced Model Size: Using techniques like separable convolution and global average pooling (GAP), the model size is reduced by 87% compared to standard CNN models while maintaining high feature extraction and learning capability.	a. The model effectively detects DoS attacks, but its ability to identify other traffic anomalies (e.g., phishing, data injection) is not addressed. Future research should evaluate its performance against various network attack types. b. The paper relies on simulation datasets (KDDcup99, NSL-KDD, UNSW-NB15) for validation. More experiments in real-world WSN environments with varied traffic patterns are needed to confirm practical applicability. c. While the model is lightweight, further optimizations may be necessary for real-time applications, given the strict latency and power consumption requirements of WSNs.
SRLGBM-DDS: An Efficient Data Balanced DoS Detection System for Wireless Sensor Networks on Big Data Environment [11].	2022	a. The STLGBM-DDS system achieves 99.95% accuracy in detecting DoS attacks on Wireless Sensor Networks (WSNs), outperforming other ML and DL models. b. It addresses data imbalance in WSN datasets using the SMOTE + Tomek-Links (STL) method, improving classification for underrepresented classes. c. The LightGBM algorithm shows superior performance in detecting WSN-specific intrusions compared to traditional methods and other hybrid DL approaches.	a. The current system focuses on the WSN-DS dataset; future work will evaluate different datasets to validate generalizability. b. The model could be improved by incorporating hybrid ML and DL techniques, such as combining LightGBM with CNN, LSTM, or GRU for better performance. c. Additional oversampling and undersampling methods will be explored to improve data balancing beyond the STL method.
A Lightweight Multilayer Machine Learning Detection	2022	a. It employs two ML models: Naïve Bayes for binary classification (First-layer) and LightGBM for multi-class classification (Second-layer).	a. The current system focuses on internal DoS attacks; future research could extend detection to

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

System for Cyber-attacks in WSN [12].		b. The system reduces communication overhead and enhances detection speed by using a mobile robot for routing critical information.	external attacks and other cyber-attacks in WSNs. b. Future work may include additional types of robots or drone-based communication systems to enhance detection speed and coverage area.
An Integrated Rule-Based and Machine Learning Technique for Efficient DoS Attack Detection in WSN [13].	2024	a. The rule-based method distinguishes normal flow from attack flow; if an attack is detected, ML models classify the specific type of DoS attack. b. The Decision Tree classifier performed best, achieving high accuracy across all attack types, while SVM showed lower performance. c. The system is fast and efficient by avoiding the use of ML models when the flow is normal, saving computational resources.	a. The proposed model could be enhanced by exploring additional lightweight classifiers for faster performance while maintaining high accuracy. b. Future work could extend the model to detect external attacks or other types of network-layer threats beyond DoS.
An Advanced Feature Selection Approach to Improve Intrusion Detection System using Machine Learning [14].	2023	a. The model combines Entropy-based Infinite Feature Selection (E-IFS) with Eigenvector Centrality, providing a refined feature set and reducing false alarms.	a. The model primarily focused on NSL-KDD and KDD-CUP99 datasets; future work could validate it on more diverse and recent datasets for generalization. b. Further research could explore integrating deep learning with the proposed feature selection technique to handle more complex attack patterns in IoT and large-scale networks.

IV. SYSTEM LAYOUT

Following is the system design for how our DoS detection model will be integrated in a WSN. It is structured in multiple layers, out of which our role will be in developing the Machine Learning Model for detecting attacks, this is shown by the Scope of our Project division in the figure. An overview of the different layers of the system is as follows–

A. *Sensor Nodes*

These are the nodes that make up the WSN, they are deployed to continuously monitor and collect data from their environment with their sensing capabilities. After collecting the data, they send it to their respective cluster heads, which aggregate the received data and send it to the base station to be used for their intended purpose. The ML model used for detection is deployed at every single one of these sensor nodes.

B. *Edge Computing Layer*

This layer processes the data collected by the sensor nodes to make it usable for detection, it consists of two parts–

1) *Data Preprocessing Unit:* This unit performs filtering of null values from the collected data
2) *Traffic Data Aggregator:* Following preprocessing, this component selects the most relevant features to be used in detective modeling. After this, the data is input to the detection model, which will use the dataset it learned from, along with the algorithm applied to it, to detect whether or not the node is being attacked.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

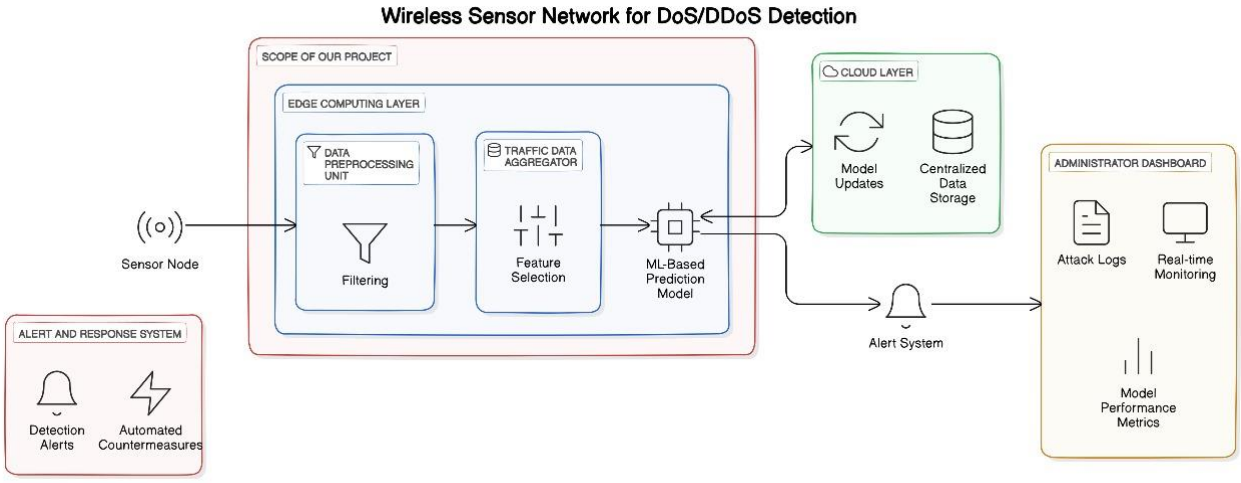

Fig. 1 System Design

C. *Cloud Layer*

If any computation if too heavy for a certain node, it sends the data forward to the cloud layer–

1) *Centralized Data Storage:* This unit contains the dataset and it also continues to store all incoming data to add to its dataset, to ensure that it learns from the incoming data as well.
2) *Model Updates:* As new data is added to the dataset, the model updates itself periodically to ensure it stays up to date.

D. *Administrator Dashboard*

This is the User interface of the part of the application that is using the WSN. It tells the user about the attacks with the help of the following units contained in it–

1) *Attack Logs:* maintains a log of all the attacks that have occurred
2) *Real-time Monitoring:* Tracks attacks as they're being detected and displays them
3) *Model Performance Metrics:* This feature provides feedback on the accuracy and efficiency of the deployed machine learning models.

E. *Alert & Response System*

The application can have a system for alerting the users in case of attacks and a predetermined response to ensure the network and data cannot be harmed further and remains safe.

V. METHODOLOGY

In this section, we will be discussing the methodology we followed for building our ML-based DoS detection model.

A. *Dataset Preparation*

The dataset that we will be using for training the model to recognize DoS attacks is the WSN-DS dataset. The classes that the data gets categorized into are–

1) *Normal:* A normal non-malicious data stream
2) *Grayhole:* The intruder node sends out Advertising Cluster Head messages (ADV-CH) with high transmission power and then drops some of the data packets when it becomes a CH.
3) *Blackhole:* The intruder node sends out Advertising Cluster Head messages (ADV-CH) with high transmission power and then drops all of the data packets when it becomes a CH.
4) *TDMA:* This attack exploits the Time Division Multiple Access protocol by acting as the CH, changing the schedule of the nodes so that all of their time slots for sending their data are the same. This causes collisions of data packets, which leads to loss of data.
5) *Flooding:* The intruder node sends out Advertising Cluster Head messages (ADV-CH) with high transmission power and then bombards the Base Station with a large volume of data [15].

As stated before, this dataset is imbalanced with respect to the number of attack instances and regular data stream instances as shown by the graph in Fig 2. According to Fig 2, there number of instances with normal data traffic is many times more than any of the instances with an attack being performed.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

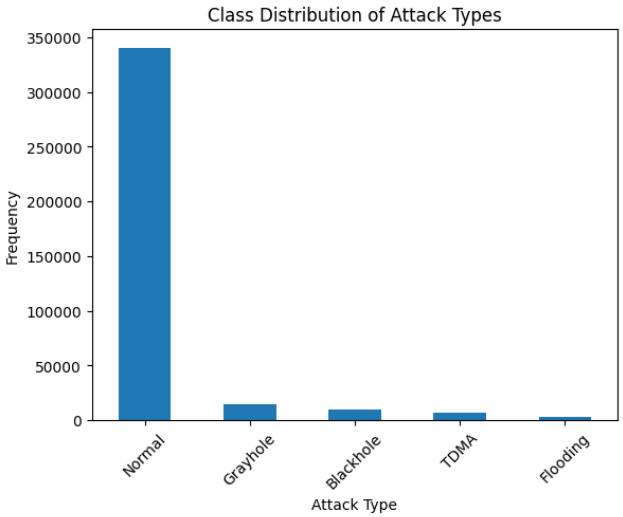

Fig. 2 Class Distribution of Attack Types Before Handling the Imbalance in Data

This imbalance is particularly problematic in machine learning models, as they tend to be biased toward the majority class. As a result, the model may achieve high overall accuracy by favoring the majority class while underperforming on the minority class. In contexts such as DoS/DDoS attack detection, this can be critical because the minority class is often the one of interest (e.g., actual attacks). If a model fails to properly learn from these minority class instances, it may miss important predictions, leading to poor performance in real-world applications where detecting the minority class is crucial [16].

To handle this imbalance, we applied Synthetic Minority Oversampling Technique, popularly known as SMOTE, the results of which are shown in Fig 3. SMOTE is a popular method used to address data imbalance by generating synthetic examples for the minority class. Unlike simple random oversampling, which duplicates existing instances, SMOTE creates new synthetic instances by interpolating between existing minority class examples. This helps introduce more diversity in the minority class without introducing duplicate data, which can lead to overfitting. By balancing the class distribution, SMOTE allows machine learning models to better learn from minority class data, improving the model's performance in identifying minority class instances and reducing bias toward the majority class. This ultimately leads to better generalization and more reliable predictions for imbalanced datasets [17].

As we can see, it made the number of instances of each class equal. This will help us to get a better performance, and the model will be able to detect attacks better as well.

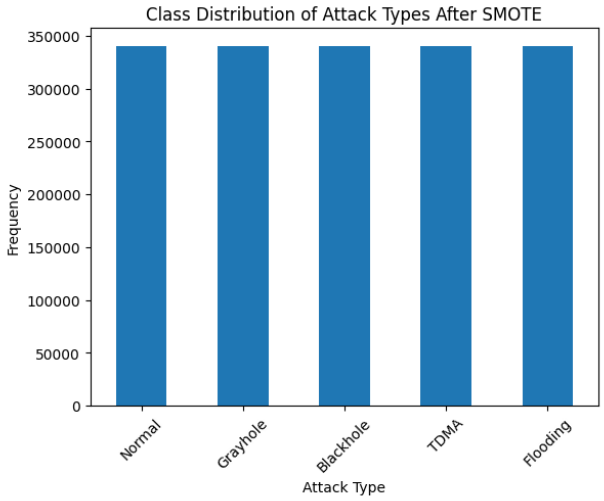

Fig. 3 Class Distribution of Attack Types After SMOTE

DOI: 10.17148/IJARCCE.2025.14444

B. *Feature Selection*

Features selection is a very critical step as was also stated in the literature survey because it determines model performance as well as interpretability. In WSN-DS, there are 18 features total but would use only relevant features for DoS attack detection. Such features comprise metrics like packet counts, transmission rates, and time intervals of different packets, which can give better insight into network behavior. Some of the feature selection techniques that performed well in recent literature are–

1) *OneR:* OneR is a straightforward algorithm that looks at each feature individually and picks the one that gives the best results for classification. It creates simple rules based on that single feature, like "if X happens, predict Y." While it's easy to use and can give surprisingly good results, it doesn't consider how features might work together, so it may not be the best choice for more complex problems [18].

2) *Gini index:* The Gini index is used in decision trees to decide the best way to split data. It helps measure how "pure" a group of items is after a split. If a group contains mostly one class, it's considered more pure. When building the tree, the algorithm looks for splits that make the groups as pure as possible, or in other words, with the lowest Gini index. This helps the tree make better decisions at each step.

3) *Information Gain:* Information gain is a feature selection technique used to measure how much a given feature contributes to reducing uncertainty (entropy) in a dataset. It's commonly applied in decision trees and other classification algorithms. The goal is to find features that provide the most informative splits of the data.

Information gain is calculated by comparing the entropy of the dataset before and after a split on a particular feature. Entropy represents the randomness or impurity in the data, and a feature with high information gain significantly reduces this entropy, indicating that it helps create clearer distinctions between classes. Features with the highest information gain are prioritized because they offer the most significant improvement in predicting the target variable, thus leading to more effective and concise models.

C. *Model Training and Testing*

Before we can start training the model with the dataset, it is divided into 3 parts at this point- Train set, Test set and Validation set. We split it in a 70:20:10 ratio. Here, three algorithms for model training have been compared.

1) Logistic Regression: This model was selected because this is a very straightforward and efficient model utilized for binary classification problems. Logistic Regression (LoR) works by estimating the probability that a given input belongs to a particular class, typically outputting a value between 0 and 1. It uses the logistic function (sigmoid function) to model the relationship between the features and the probability of the outcome, making it ideal for predicting categorical outcomes. Since it assumes a linear relationship between the input features and the log-odds of the output, it is computationally less expensive and easy to interpret, making it a go-to choice for many binary classification tasks [19].

2) Decision Tree: A Decision Tree is a versatile and intuitive model used for both classification and regression tasks. It works by recursively splitting the dataset into subsets based on feature values that result in the most homogenous groups, typically measured using criteria like Gini index or information gain. At each node, the feature that best separates the data is chosen, and the process continues until the tree reaches a leaf node, representing a decision or prediction. Decision Trees are easy to understand and interpret, as they mimic human decision-making by following simple if-then rules. However, they can be prone to overfitting, especially with deep trees, but techniques like pruning can help reduce this risk, improving the model's generalization [20].

3) REPTree: REPTree (Reduced Error Pruning Tree) is a decision tree algorithm designed for efficiency and accuracy in regression and classification tasks. It builds a tree by splitting data to minimize impurity measures and then applies reduced error pruning to improve generalization by removing non-essential branches. Its main advantages include speed and reduced overfitting due to pruning, making it well-suited for large datasets. However, like all decision trees, it can be biased towards classes with more instances if not managed properly [21].

Then, the portion of the dataset was used to train each model, while the rest was used as a test: for instance, 70% for training and the other 30% for testing. Cross-validation methodologies were utilized to ensure that the model performance stayed robust.

D. *Validation*

Strict validation accuracy of the models was verified by the application of K-Fold Cross-Validation. K-Fold Cross-Validation is the procedure in which the dataset is split into K folds. At each iteration, one fold is treated as the validation set whereas the remaining K-1 folds are used to train that model. This procedure is repeated K times for each fold to be treated as the validation set once.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

It helps to reduce some overfitting problems and gives a more precise accuracy measure of what the model would produce at unseen data. The totals of the results from all the folds are then used to get further performance metrics, such as accuracy, precision, recall, and the F1-score. Therefore, this strong validation method ensures that the selected model generally applies and remains predictive on different subsets of data, making the DoS detection system more reliable.

E. *Performance Evaluation*

The performance of each model is evaluated with such metrics as accuracy, precision, recall, F1-score. To plot the performance of the models in multiclass classification, a confusion matrix has been adopted. These metrics were very crucial for the system in assessing the capability of the models to differentiate various DoS attacks detected within the WSN. These metrics are given by–

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

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

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

4) $F1 - score = \frac{2 \times Precision \times Recall}{Precision+Recall}$

VI. RESULTS

We tried three machine learning models in our experiment for Denial of Service in Wireless Sensor Networks, these were, Logistic Regression, Decision Tree, and REPTree. The results showed that the better-performing one among the three models was indeed the REPTree model in respect to accuracy, precision, recall, and F1-score. Here, while REPTree was 99.95% accurate, Logistic Regression and Decision Tree mentioned accuracies of 89.90% and 99.94%, respectively.

This is clear from the improvement in how REPTree performs its tasks as it is able to handle complex data distributions in relation to keeping a tight control over the class imbalance observed within the dataset. The results show the efficiency of the REPTree model in correctly identifying both types of DoS attacks, thus suitable for real-time deployment in WSNs.

TABLE III RESULTS

Model	Accuracy (%)	Blackhole F1-score	Flooding F1-score	Grayhole F1-score	Normal F1-score	TDMA F1-score
Logistic Regression	89.903	0.83	1.000	0.73	0.96	0.96
Decision Tree	99.945	1.000	1.000	1.000	1.000	1.000
REPTree	99.949	1.000	1.000	1.000	1.000	1.000
Logistic Regression + PCA	70.737	0.72	0.87	0.51	0.71	0.66
Decision Tree + PCA	99.909	1.000	1.000	1.000	1.000	1.000
REPTree + PCA	99.915	1.000	1.000	1.000	1.000	1.000

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

VII. CONCLUSION

This paper succeeded in developing an attack detection system of DoS attacks in WSN using machine learning with the WSN-DS dataset. Using SMOTE technique, the model capability to classify attacks has improved between classes. Evaluation of Logistic Regression, Decision Tree, and REPTree proved that REPTree can be more reliable for model building, with respect to superb accuracy and strong anomalous detection capabilities in network traffic.

Our findings show plenty of importance of machine learning methods toward the upgradation of security mechanisms of WSNs against DoS attacks and open ways and avenues for further research in a variety of sophisticated algorithms and hybrid mechanisms for raising detection capabilities to even higher levels.

REFERENCES

[1]. Zijie, F., Al-Shareeda, M. A., Saare, M. A., Manickam, S., & Karuppayah, S. (2023). Wireless sensor networks in the internet of things: review, techniques, challenges, and future directions. Indonesian Journal of Electrical Engineering and Computer Science, 31(2), 1190. https://doi.org/10.11591/ijeecs.v31.i2.pp1190-1200

[2]. P. Singh, D. O. Gupta, and S. Saini, "A Brief Research Study of Wireless Sensor Network", 2017.

[3]. Pathan, A., Lee, N. H., & Hong, N. C. S. (2006). Security in wireless sensor networks: issues and challenges. 2006 8th International Conference Advanced Communication Technology. https://doi.org/10.1109/icact.2006.206151

[4]. Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys & Tutorials, 10(4), 56–76. https://doi.org/10.1109/surv.2008.080406

[5]. X. Cao, L. Liu, Y. Cheng and X. Shen, "Towards Energy-Efficient Wireless Networking in the Big Data Era: A Survey," in IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 303-332, Firstquarter 2018, doi: 10.1109/COMST.2017.2771534

[6]. Quincozes, S. E., Kazienko, J. F., & Quincozes, V. E. (2023). An extended evaluation on machine learning techniques for Denial-of-Service detection in Wireless Sensor Networks. Internet of Things, 22, 100684. https://doi.org/10.1016/j.iot.2023.100684

[7]. Suresh, B., & Prasad, G. S. C. (2023). An Energy Efficient Secure routing Scheme using LEACH protocol in WSN for IoT networks. Measurement Sensors, 30, 100883. https://doi.org/10.1016/j.measen.2023.100883

[8]. Elsadig, M. A. (2023). Detection of Denial-of-Service Attack in Wireless Sensor Networks: A lightweight machine learning approach. IEEE Access, 11, 83537–83552. https://doi.org/10.1109/access.2023.3303113.

[9]. Ahmad, R., Wazirali, R., Bsoul, Q., Abu-Ain, T., & Abu-Ain, W. (2021). Feature-Selection and Mutual-Clustering approaches to improve DOS detection and maintain WSNs' lifetime. Sensors, 21(14), 4821. https://doi.org/10.3390/s21144821.

[10]. C. Yao, Y. Yang, K. Yin and J. Yang, "Traffic Anomaly Detection in Wireless Sensor Networks Based on Principal Component Analysis and Deep Convolution Neural Network," in IEEE Access, vol. 10, pp. 103136-103149, 2022, doi: 10.1109/ACCESS.2022.3210189

[11]. M. Dener, S. Al and A. Orman, "STLGBM-DDS: An Efficient Data Balanced DoS Detection System for Wireless Sensor Networks on Big Data Environment," in IEEE Access, vol. 10, pp. 92931-92945, 2022, doi: 10.1109/ACCESS.2022.3202807

[12]. S. Ismail, D. Dawoud and H. Reza, "A Lightweight Multilayer Machine Learning Detection System for Cyber-attacks in WSN," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2022, pp. 0481-0486, doi: 10.1109/CCWC54503.2022.9720891

[13]. Swami, S., Singh, P., & Chauhan, S. S. (2024). An Integrated Rule-Based and Machine Learning Technique for Efficient DoS Attack Detection in WSN. 2024 2nd International Conference on Disruptive Technologies (ICDT). https://doi.org/10.1109/icdt61202.2024.10489560

[14]. Kaur, N., Singla, J., Mathur, G., Talwani, S., & Malik, N. (2023). An Advanced Feature Selection Approach to Improve Intrusion Detection System using Machine Learning. Proceedings of the Seventh International Conference on Electronics, Communication and Aerospace Technology (ICECA 2023). https://doi.org/10.1109/iceca58529.2023.10394718

[15]. Almomani, I., Al-Kasasbeh, B., & Al-Akhras, M. (2016). WSN-DS: a dataset for intrusion detection systems in wireless sensor networks. Journal of Sensors, 2016, 1–16. https://doi.org/10.1155/2016/4731953

[16]. K. Oksuz, B. C. Cam, S. Kalkan and E. Akbas, "Imbalance Problems in Object Detection: A Review," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3388-3415, 1 Oct. 2021, doi: 10.1109/TPAMI.2020.2981890

[17]. Elreedy, D., & Atiya, A. F. (2019). A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. Information Sciences, 505, 32–64. https://doi.org/10.1016/j.ins.2019.07.070

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14444

[18]. Holte, R.C. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning 11, 63–90 (1993). https://doi.org/10.1023/A:102263111893

[19]. Couronné, R., Probst, P., & Boulesteix, A.-L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. BMC Bioinformatics, 19(1). doi:10.1186/s12859-018-2264-5

[20]. Rokach, L. (2016). Decision forest: Twenty years of research. Information Fusion, 27, 111–125. doi:10.1016/j.inffus.2015.06.005

[21]. Esposito, F., Malerba, D., Semeraro, G., & Kay, J. (1997). A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5), 476–493. doi:10.1109/34.589207