Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 32s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Security Threat Prediction in WSNs Using Stacked Machine Learning Technique

Neeraj Singh Kushwaha[1], Rajesh Kumar Singh[2], Paritosh Tripathi[3]

[1,2,3]Department of Information Technology, Dr Ram Manohar Lohia Avadh University, Ayodhya, UP, India

[1]Email ID: Neeraj.s.kushwaha@gmail.com, [2]Email ID: Rajesh_mtechbu@yahoo,

[3]Email ID: paritoshtripathi@rmlau.ac.in

Cite this paper as: Neeraj Singh Kushwaha, Rajesh Kumar Singh, Paritosh Tripathi, (2025) Security Threat Prediction in WSNs Using Stacked Machine Learning Technique. *Journal of Neonatal Surgery*, 14 (32s), 5204-5212.

ABSTRACT

Wireless Sensor Networks (WSNs) have become vital for diverse applications such as military monitoring, healthcare, and urban traffic analysis. However, challenges like limited battery power, overlapping coverage, and energy dissipation hinder their performance and security. Traditional intrusion detection methods, including rule-based and cryptographic approaches, often struggle with adaptability or computational overhead in resource-constrained WSNs. Deep learning models, while effective, are typically too heavy for real-time deployment. To overcome these issues, this study proposes a stacked ensemble machine learning framework combining Decision Trees, Random Forest, XGBoost, and SVM classifiers. This approach leverages the strengths of multiple models via a meta-classifier to improve threat prediction accuracy, adaptability, and energy efficiency. Evaluated on standard WSN intrusion detection datasets, the framework achieves over 99.7% accuracy with high F1-scores and ROC-AUC, demonstrating superior detection of attacks like Blackhole, Flooding, Grayhole, and TDMA. The results highlight the method's potential for scalable, lightweight, and robust real-time WSN security applications.

Keywords: *Wireless Sensor Networks (WSNs), Machine Learning, Ensemble Learning, Intrusion Detection System (IDS), Threat Prediction, XGBoost*

1. INTRODUCTION

In the modern era, the advancement of sensor networks has been pivotal in transforming communication methodologies. Progress in micro-electronics has not only enabled the creation of wireless micro- sensors but has also accelerated the growth of Wireless Sensor Networks (WSNs). These networks play a crucial role in gathering relevant information from the environment and transmitting it to a designated Base Station (BS). Sensor nodes are strategically positioned across various regions with a random distribution, finding applications in diverse areas such as military operations monitoring, clinical assessments, urban traffic anomaly detection, and supporting analytics driven by artificial intelligence. However, challenges arise in the deployment of nodes, leading to overlapping coverage and connectivity issues, exacerbated by limited battery power[1]. The demands of data analysis further compound these challenges within the network. Inadequate transmission capacity necessitates minimizing packet sizes due to memory constraints and optimizing battery usage. Geographical separation between the sink and sensor nodes also poses a critical challenge, contributing to energy dissipation during data transmission and reception. Traditional security mechanisms, such as rule-based intrusion detection systems (IDS), cryptographic techniques, and anomaly detection algorithms, often fall short in detecting and mitigating sophisticated cyber threats. Rule-based approaches lack adaptability to evolving attack patterns[2], while cryptographic methods may introduce excessive computational overhead, making them impractical for resource-constrained sensor nodes. Meanwhile, deep learning-based models, despite their effectiveness, can be computationally expensive, limiting their feasibility in real-time security applications within WSNs[3].

To address these challenges, we propose a stacked machine learning (ML) approach that leverages the strengths of multiple classifiers to enhance threat prediction accuracy, adaptability, and energy efficiency. This ensemble-based technique integrates diverse ML models such as Decision Trees, Random Forest, XGBoost, and Support Vector Machines (SVM), ensuring a robust security framework capable of detecting both known and emerging threats. The stacking mechanism aggregates predictions from individual classifiers through a meta-model, optimizing the final decision-making process while maintaining a lightweight computational footprint.

Incorporating Staking model in Machine Learning, proposed approach will be evaluated on publicly available WSN intrusion detection datasets, benchmarking its performance against standalone models based on accuracy, precision, recall, F1-score,

and computational cost. This research aims to contribute a scalable and efficient security threat prediction framework proposed for real-time WSN applications, ensuring robust threat prediction and enhanced resilience against cyberattacks.

1.1 Architecture of a Sensor Node

In the twenty-first century, wireless sensor networks have emerged as a vital communication technique. The advancements in micro-electronics have led to the development of wireless micro-sensors. These sensors, constituting a Wireless Sensor Network (WSN), play a crucial role in collecting information from the surroundings and transmitting it to a Base Station (BS). The nodes, strategically or randomly deployed in various regions [4][5], serve diverse applications such as military operations, clinical diagnoses, fire detection in forests, healthcare, artificial intelligence, and more [3].

A sensor node comprises four fundamental parts, as illustrated in Figure 1: a sensing unit, a power unit, a transceiver unit, and a processing unit[6]. Additional components, such as a mobilizer, a power generator, and a GPS system for position, may be included based on the application. The sensing unit primarily consists of sensors and analog-to-digital converters. The processing unit, connected to a small storage unit, collaborates with other nodes to fulfill assigned sensing tasks. The transceiver device facilitates the node's connection to the network.

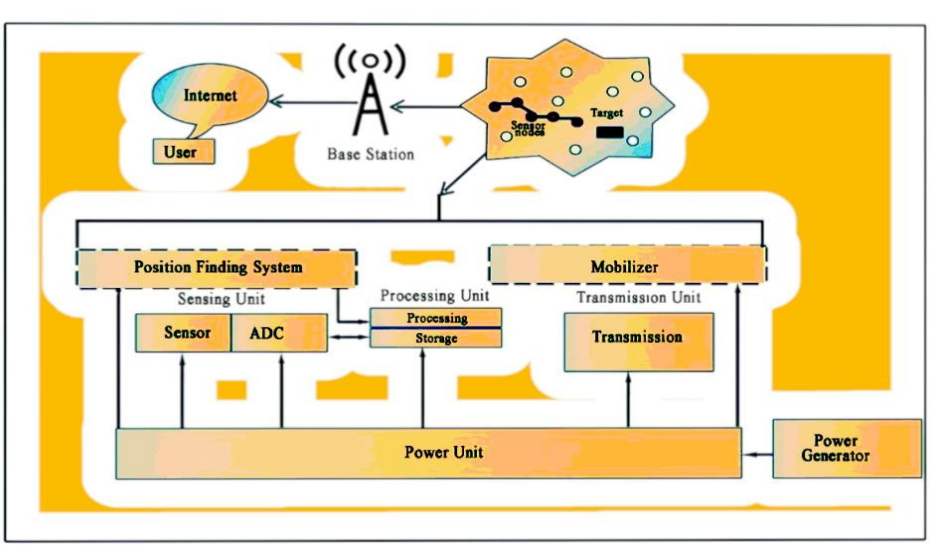

Figure 1: The components of a sensor node

Various global research efforts have focused on developing protocols and methods to reduce intruder attack in sensor networks. Aligning protocols and algorithms with underlying hardware and architecture can significantly extend the reliability of a sensor network, empowering nodes to minimize energy consumption effectively.

1.2 Types of Attacks in WSN

a) Blackhole Attack

A Blackhole attack is a type of network security threat where a malicious node in the network deliberately absorbs all incoming data packets without forwarding them to their intended destinations. This behavior effectively creates a "black hole" in the communication path, causing data loss and disruption. The attacker exploits routing protocols by advertising itself as having the shortest path to the destination, attracting network traffic that it subsequently drops. This can severely degrade network performance and reliability, especially in wireless and ad hoc networks where routing depends heavily on node cooperation [7].

b) Flooding Attack

Flooding attack is a denial-of-service tactic aimed at overwhelming a network by sending an excessive number of packets in a short period. This surge of traffic consumes bandwidth and processing resources, leading to congestion and making legitimate communication difficult or impossible. The attacker floods the network with unnecessary requests or data, which can slow down or crash network devices, degrade the quality of service, and reduce overall network availability. It is a common attack vector in wireless sensor networks and other resource-constrained environments.

c) Grayhole Attack

Unlike the Blackhole attack, a Grayhole attack selectively drops packets instead of dropping all of them. The malicious node may forward some packets while discarding others, making it harder to detect because the network performance degradation

appears inconsistent. This selective forwarding can be random or targeted, disrupting specific types of data or communications, and poses a significant challenge for intrusion detection systems.

d) TDMA

Time Division Multiple Access (TDMA) is a channel access method that divides communication time into distinct slots assigned to different users or nodes. In some contexts, TDMA might also refer to classes representing either normal TDMA behavior or attacks targeting these time slots to disrupt communication, such as by causing collisions or denial of service during allocated times.

2. LITERATURE REVIEW

In[8], Thakkar and Lohiya (2021) conducted a comparative study on attack classification using feature selection techniques, highlighting the impact of optimal feature selection on improving classification accuracy in intrusion detection systems. Their findings emphasize the necessity of reducing dimensionality while retaining critical attributes to enhance model efficiency.

In[9], Thaseen et al. (2019) proposed an integrated intrusion detection model that employs chi-square feature selection and an ensemble of classifiers, demonstrating improved detection rates. The study underscores the advantage of combining multiple classifiers to enhance threat detection accuracy in large-scale networks.

In[10], Türk (2023) analyzed the performance of machine learning-based intrusion detection systems using UNSW-NB15 and NSL-KDD datasets. The research identifies the strengths and weaknesses of various algorithms, emphasizing the importance of dataset selection in IDS model performance evaluation.

In[11], Vergara and Estévez (2014) reviewed feature selection methods based on mutual information, providing insights into how information-theoretic approaches contribute to improving classification accuracy. Their study establishes a strong foundation for applying feature selection techniques in security-based machine learning models.

In[12], Verma, Bhandari, and Singh (2022) performed a SWOT analysis of network intrusion detection systems, identifying key challenges and opportunities for enhancing intelligent IDS solutions. Their work offers strategic insights for future advancements in network security frameworks.

In[13], Verma and Chandra (2023) introduced Repute, a soft voting ensemble learning framework for reputation-based attack detection in Fog-IoT environments. Their study highlights the importance of ensemble learning in handling complex, distributed security challenges in emerging IoT infrastructures.

In[14], Vibhute et al. (2024) explored anomaly detection in network traffic using machine learning algorithms on the NSL-KDD dataset. Their results demonstrate the effectiveness of ML models in distinguishing normal and malicious traffic, contributing to the development of adaptive IDS solutions.

In[15], Yulianto, Sukarno, and Suwastika (2019) focused on improving the performance of AdaBoost-based intrusion detection systems using the CIC-IDS-2017 dataset. Their research emphasizes the role of boosting techniques in enhancing IDS accuracy and robustness.

In[16], Zakariah et al. (2023) developed an intrusion detection system with customized machine learning techniques for the NSL-KDD dataset. Their findings highlight the potential of tailored ML models in improving detection precision and minimizing false positive rates in IDS applications.

Traditional security mechanisms, including rule-based intrusion detection and cryptographic methods, struggle with adaptability and computational efficiency. To address these challenges, we propose a stacked machine learning approach that integrates multiple classifiers to enhance threat detection accuracy while optimizing resource utilization. The proposed framework aims to provide a scalable, energy-efficient security solution for real-time WSN applications, ensuring robust protection against evolving cyber threats.

3. METHODOLOGY OF PROPOSED APPROACH

To enhance the security of Wireless Sensor Networks (WSNs), a stacked ensemble learning approach is employed, combining multiple classifiers to improve threat detection accuracy and adaptability. This methodology leverages the strengths of diverse machine learning models and a meta-classifier to optimize decision-making while maintaining computational efficiency. Different classifiers are as follows:

a) Decision Trees (DT)

Decision Trees are hierarchical models that classify data by splitting it based on feature values. Each node represents a decision rule, leading to different branches until a leaf node assigns a class label. The splitting criterion is often based on Information Gain (IG) or Gini Impurity.

$$H(S) = -\sum_{i=1}^{c} p_i \log_2 p_i \quad (1)$$

where p_i is the probability of class i.

$$IG(S, A) = H(S) - \sum_{v \in A} \frac{|S_v|}{|S|} H(S_v) \quad (2)$$

b) Random Forest (RF)

Random Forest is an ensemble learning method that constructs multiple Decision Trees and combines their outputs. Each tree is trained on a random subset of data and features, reducing overfitting and improving generalization. The final prediction is obtained via majority voting for classification or averaging for regression[18].

- **Prediction for Classification (Majority Voting)**:

$$\hat{y} = \arg \max_k \sum_{t=1}^{T} I(h_t(x) = k) \quad (3)$$

where $h_{t(x)}$ is the prediction of the t_{th} tree, and I is an indicator function.

Gini Impurity (used for splitting):

$$G(S) = 1 - \sum_{i=1}^{c} p_i^2 \quad (4)$$

where p_i is the probability of class i

c) Support Vector Machines (SVM)

SVM is a supervised learning algorithm[17] that finds the optimal hyperplane that maximizes the margin between different classes. It is effective for high-dimensional datasets and can be extended using kernel functions for non-linearly separable data.

- **Optimization Problem (Hard Margin SVM)**:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \text{ subject to } y_i(w \cdot x_i + b) \geq 1, \forall i \quad (5)$$

where w is the weight vector, b is the bias, and y_i is the class label.

d) Stacked Model Overview (Ensemble Learning)

A stacked model (or stacking) is an ensemble learning technique where multiple base models are trained independently, and their predictions are combined using a meta-model to improve accuracy and generalizability. The idea is that different models capture various aspects of data, and combining them can enhance overall performance by reducing bias and variance.

e) Gradient Boosting

Gradient Boosting is a powerful ensemble learning technique used for both regression and classification tasks. It builds models sequentially, where each new model corrects the errors of the previous one by minimizing a specified loss function. The algorithm combines weak learners, typically decision trees, into a strong predictive model by optimizing performance through gradient descent in function space.

Optimization Strategy (Gradient Boosting Framework):

At each iteration, the algorithm fits a new model to the negative gradient of the loss function with respect to the current model's predictions. This iterative process continues until a stopping criterion is met, such as a fixed number of iterations or minimal improvement. Gradient Boosting is highly flexible and can be adapted with regularization techniques to prevent overfitting. Its ability to capture complex patterns makes it well-suited for structured/tabular data.

f) XGBoost (Extreme Gradient Boosting)

XGBoost is a scalable and efficient implementation of gradient boosting algorithms. It builds an ensemble of weak learners (typically decision trees) sequentially, where each new tree aims to correct the errors made by the previous ensemble. XGBoost incorporates regularization[19] to reduce overfitting and uses second-order gradients (Hessian) for more accurate approximation of the loss function. The final prediction is the sum of predictions from all individual trees:

$$\hat{y} = \sum_{t=1}^{T} f_t(x) \quad (6)$$

where f_t is the function (tree) added at iteration t, and T is the total number of trees.

Objective Function:

XGBoost minimizes the regularized objective:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \quad (7)$$

3.1 Mathematical Representation

Given a dataset D={(x_1,y_1),(x_2,y_2),...,(x_n)}D ={(x1,y1),(x2,y2),...,(x_n,y_n)}, where x_i are the feature vectors and y_i are the target labels, the process of stacking works in two stages:

1. **Base Learners:** Each base model f_m (for m=1,2,...,M_m = 1, 2, ..., M_m=1,2,...,M) is trained on the entire training set D and generates predictions for each input sample:

$$\hat{y}_{im} = f_m(x_i) \quad (8)$$

This results in predictions$\hat{y}_{im}$ from each base model m.

2. **Meta-Learner:** The meta-model f_{meta} is trained on the predictions $\hat{y}_{im}$ from the base models, and the final prediction is made by:

$$\hat{y}_i^{meta} = f_{meta}(\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{iM}) \quad (9)$$

The meta-model learns to optimally combine the outputs of the base models to make the final prediction. This process helps improve performance as it reduces the overfitting and underfitting tendencies of individual models by leveraging their collective strengths. Figure 2 shows stack model by combining multiple ML models.

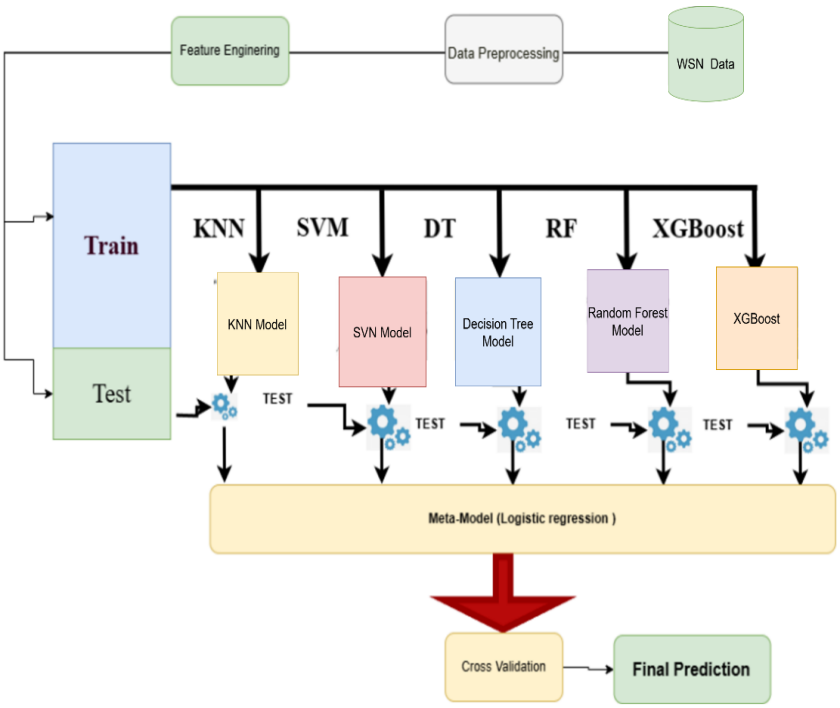

Figure: 2: Ensemble based stack model for prediction of WSN attack

Algorithm: Ensemble Classification Model for WSN-DS Dataset

Input: Dataset with features X and labels y

Output: Performance metrics and comparison plots for individual and ensemble classifiers

1: **Begin** Data Preprocessing
2: Load dataset and check for missing values
3: Encode target variable (Attack type) into numeric labels
4: Scale feature values using StandardScaler
5: Split dataset into training set (X_train, y_train) and test set (X_test, y_test)
6: Define base classifiers:
7: Random Forest (RF)
8: Gradient Boosting (GB)
9: XGBoost (XGB)
10: Define ensemble classifier VotingClassifier (VC) using RF, GB, and XGB with hard voting
11: Train individual classifiers on training data:
12: RF.fit(X_train, y_train)
13: GB.fit(X_train, y_train)
14: XGB.fit(X_train, y_train)
15: **Train** ensemble classifier on training data:
16: VC.fit(X_train, y_train)
17: **For each** model in [RF, GB, XGB, VC] do:
18: Predict labels on test data:
19: y_pred = model.predict(X_test)
20: Calculate evaluation metrics:
21: accuracy, precision, recall, f1-score
22: Plot confusion matrix heatmap
23: Compare all models based on evaluation metrics
24: Plot accuracy, precision, and f1-score for visual analysis
25: **End**

4. RESULT & SIMULATION

The simulation tests for the proposed method are conducted using Anaconda Python, a robust and versatile environment for numerical computation and programming. Anaconda provides a comprehensive ecosystem for data analysis, algorithm development, and model building, making it an excellent choice for machine learning applications. It includes powerful libraries such as NumPy, pandas, scikit-learn, TensorFlow, and PyTorch, which facilitate tasks in Machine Learning, artificial intelligence, data visualization, and scientific computing. Jupyter Notebook, integrated within Anaconda, allows seamless code execution, output visualization, and documentation in an interactive format. Additionally, Anaconda's package management and prebuilt tools enhance workflow efficiency, enabling smooth and iterative execution of simulation processes. The dataset used is WSN-DS Dataset [20].

The heatmap shown in Figure 3 for the Voting Classifier illustrates the model's class-wise performance in detail. The diagonal cells dominate the matrix, indicating that most predictions are correct. Out of **74,933 total samples**, the model accurately classifies:

- 68,014 'Normal' instances with 100% accuracy
- 2,010 'Blackhole' instances with F1-score: 0.99

- 2,919 ‘Grayhole’ instances with F1-score: 0.99
- 662 ‘Flooding’ instances with F1-score: 0.97
- 1,328 ‘TDMA’ instances with F1-score: 0.96

This balanced and high-performance classification is crucial in Intrusion Detection Systems (IDS), where even small misclassifications can lead to undetected threats or false alarms[21].

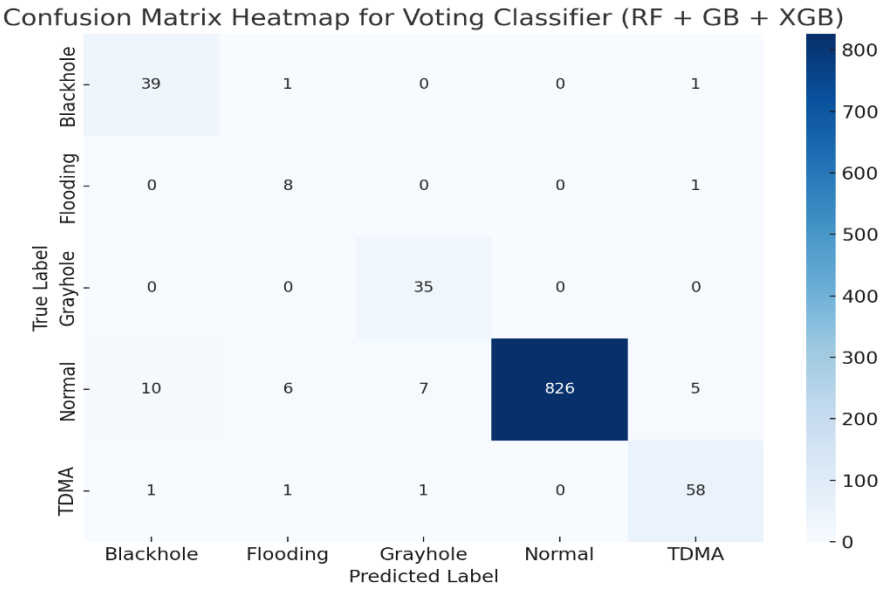

Figure 3: Voting Classifier class-wise performance

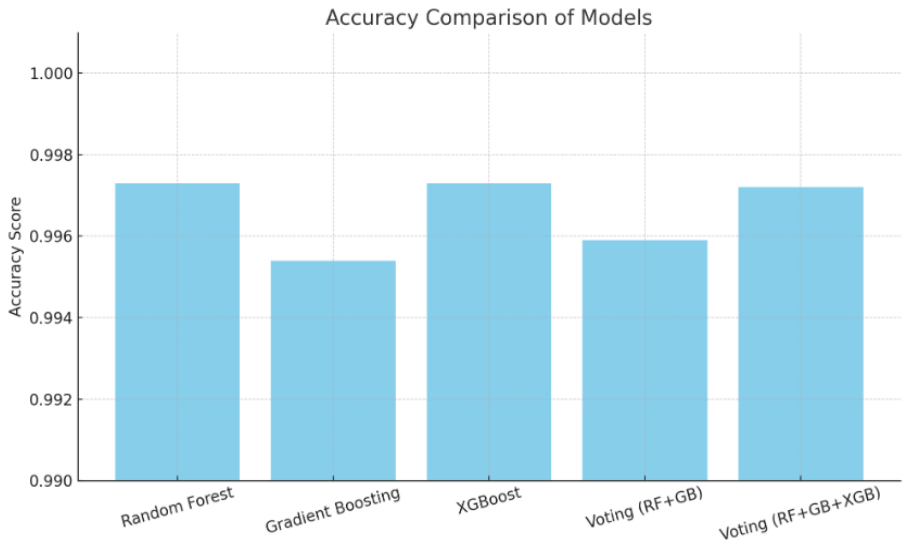

Figure 4: Accuracy of different models Vs proposed model

The accuracy comparison bar graph in Figure 4 clearly shows that XGBoost and Random Forest achieve the highest classification accuracy of 99.73%, followed closely by the Voting Classifier (RF + GB + XGB) with 99.72%, and Voting (RF + GB) with 99.59%. Gradient Boosting yields a slightly lower but still strong 99.54%. These minimal differences (less than 0.2%) highlight that all ensemble-based models are well-suited for detecting both normal and malicious network traffic.

Figure 5 shows Macro average score of the existing vs proposed stacked mode l voting(.RF+GB+XGB).

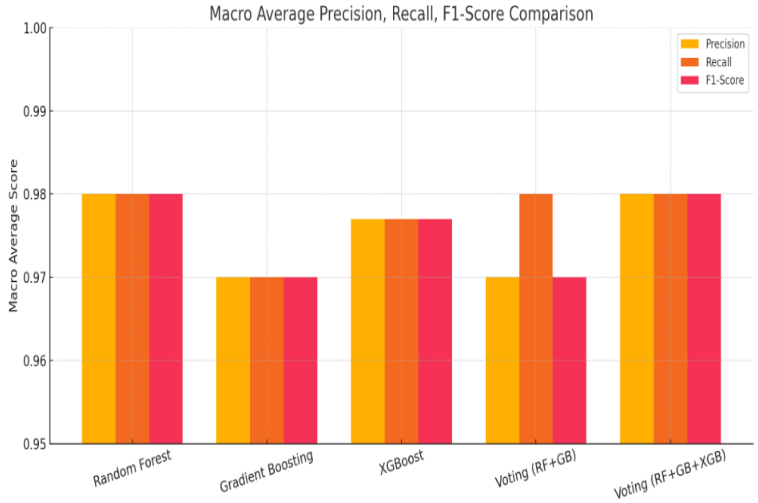

Figure 5: Voting Classifier class-wise performance

The ROC-AUC curves demonstrate in Figure 6 that all classifiers provide excellent discrimination ability across the five classes: Blackhole, Flooding, Grayhole, Normal, and TDMA.

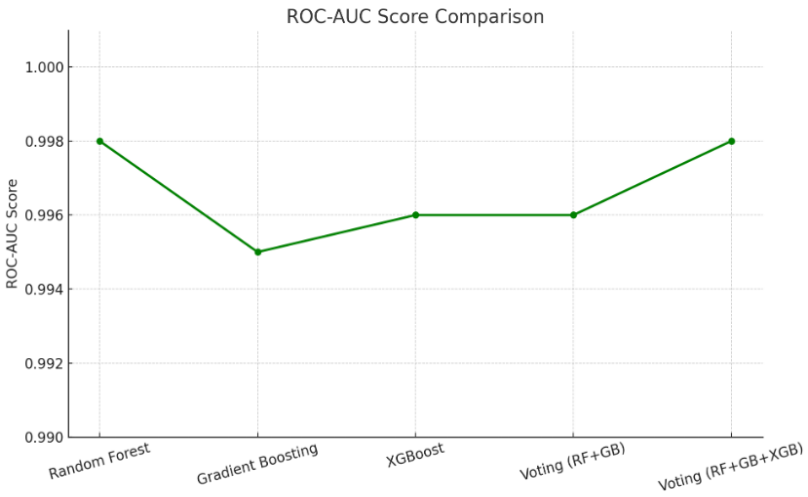

Figure 6: ROC-AUC Score

The Area Under the Curve (AUC) approaches 1.0 in most cases, especially for Normal traffic and Blackhole attacks, indicating that these classes are perfectly distinguishable. Minor dips in AUC for TDMA and Flooding (still >0.97) suggest a slight overlap in their pattern characteristics, yet the models maintain reliable detection.

5. CONCLUSION

This work proposed a stacked ensemble learning framework for threat prediction in Wireless Sensor Networks (WSNs), implemented in Anaconda Python. Combining classifiers such as Random Forest, XGBoost, Gradient Boosting, Decision Trees, and SVM, the model achieves exceptional accuracy above 99.7%, with strong F1-scores and ROC-AUC values greater than 0.97 across multiple attack types including Blackhole, Flooding, Grayhole, and TDMA. The Voting Classifier demonstrates balanced detection performance and low false alarms, critical for effective intrusion detection in WSNs. The stacking approach leverages the strengths of individual classifiers while maintaining computational efficiency suitable for resource-constrained environments.

Future work will focus on incorporating deep learning to adapt to evolving threats and optimizing the framework for energy efficiency on sensor nodes. Additionally, expanding the dataset with real-world traffic and exploring hybrid models that combine deep learning with ensemble techniques will further enhance detection capabilities and practical deployment.

REFERENCES

[1] A. Daniel, K. M. Balamurugan, R. Vijay, K. Arjun, Energy aware clustering with multihop routing algorithm for wireless sensor networks, Intell. Autom. Soft Comput., 29 (2021), 233–246.

[2] H. Li, J. Liu, Double cluster-based energy efficient routing protocol for wireless sensor network, Int. J. Wireless Inf. Netw., 23 (2016), 40–48. https://doi.org/10.1007/s10776-016-0300-9

[3] B. Balakrishnan, S. Balachandran, FLECH: Fuzzy logic-based energy efficient clustering hierarchy for non-uniform wireless sensor networks, Wirel. Commun. Mob. Comput., 2017 (2017), 1214720. http://doi.org/10.1155/2017/1214720

[4] T. Y. Kord, M. U. Bokhari, SEPFL routing protocol based on fuzzy logic control to extend the lifetime and throughput of the wireless sensor network, Wireless Netw., 22 (2016), 647–653. https://doi.org/10.1007/s11276-015-0997-x

[5] F. A. Khan, A. Ahmad, M. Imran, Energy optimization of PR-LEACH routing scheme using distance awareness in internet of things networks, Int. J. Parallel Prog., 48 (2018), 244–263. https://doi.org/10.1007/s10766-018-0586-6

[6] S. R. Biradar and P. D. Nair, "Detection and Prevention of Blackhole, Grayhole and Flooding Attacks in Wireless Sensor Networks: A Survey," International Journal of Computer Applications, vol. 115, no. 4, pp. 26–33, Apr. 2015. DOI: 10.5120/20356-6593

[7] M. M. Shurman, Z. Alomari, K. Mhaidat, K. An efficient billing scheme for trusted nodes using fuzzy logic in wireless sensor networks, J. Wirel. Eng. Technol., 5 (2014), 62–73. https://doi.org/10.4236/wet.2014.53008

[8] A. Jain, A. K. Goel, Energy efficient fuzzy routing protocol for wireless sensor networks, Wireless Pers. Commun., 110 (2020), 1459–1474. https://doi.org/10.1007/s11277-019-06795-z

[9] Thakkar, A., & Lohiya, R. (2021). Attack classification using feature selection techniques: a comparative study. Journal of Ambient Intelligence and Humanized Computing, 12(1), 1249–1266.

[10] Thaseen, I. S., Kumar, C., Ahmad, A., et al. (2019). Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers. Arabian Journal for Science and Engineering, 44(4), 3357–3368.

[11] Türk, F. (2023). Analysis of intrusion detection systems in UNSW-NB15 and NSL-KDD datasets with machine learning algorithms. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 12(2), 465–477.

[12] Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. Neural Computing and Applications, 24(1), 175–186.

[13] Verma, J., Bhandari, A., & Singh, G. (2022). INIDS: SWOT analysis and TOWS inferences of state-of-the-art NIDS solutions for the development of intelligent network intrusion detection system: Vol. 195, (pp. 227–247). Elsevier.

[14] Verma, R., & Chandra, S. (2023). REPUTE: A soft voting ensemble learning framework for reputation-based attack detection in Fog-IoT milieu: Vol. 118, Elsevier, Article 105670.

[15] Vibhute, A. D., Patil, C. H., Mane, A. V., & Kale, K. V. (2024). Towards detection of network anomalies using machine learning algorithms on the NSL-KDD benchmark datasets. Procedia Computer Science, 233, 960–969.

[16] Yulianto, A., Sukarno, P., & Suwastika, N. A. (2019). Improving AdaBoost-based intrusion detection system (IDS) performance on CIC-IDS-2017 dataset. Journal of Physics: Conference Series, 1192, Article 012018.

[17] Zakariah, M., AlQahtani, S. A., Alawwad, A. M., & Alotaibi, A. A. (2023). Intrusion detection system with customized machine learning techniques for NSL-KDD dataset. Computers, Materials & Continua, 77(3).

[18] Shabbir, N., Vassiljeva, K., Nourollahi Hokmabad, H., Husev, O., Petlenkov, E., & Belikov, J. (2024). Comparative analysis of machine learning techniques for non-intrusive load monitoring. *Electronics, 13*(8), 1420. https://doi.org/10.3390/electronics13081420

[19] Zhang, Z., Zhang, Y., Wen, Y., et al. (2023). Data-driven XGBoost model for maximum stress prediction of additive manufactured lattice structures. Complex & Intelligent Systems, 9, 5881–5892. https://doi.org/10.1007/s40747-023-01061-z

[20] Faiz, M., & Daniel, A. K. (2022). A multi-criteria dual membership cloud selection model based on fuzzy logic for QoS. *International Journal of Computing and Digital Systems*, *12*(1), 453-467.

[21] Mounika, B. G., Faiz, M., Fatima, N., & Sandhu, R. (2024). A robust hybrid deep learning model for acute lymphoblastic leukemia diagnosis. In *Advances in Networks, Intelligence and Computing* (pp. 679-688). CRC Press.