Smart Intrusion Detection in Wireless Sensor Networks Using Optimized Ensemble Learning

K S Lokesh[1], Dunna Nikitha Rao2, C Likhitha[3]

[1]P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: amloki.2003@gmail.com, ORC-ID: https://orcid.org/0009-0003-9558-2348

[2]Academic Consultant, Sri Padmavati Mahila Visvavidyalayam, Tirupati, E-mail: rajnikki8195@gmail.com

[3]Assistant Professor, Department of CSE(AI & ML), Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: likhithamandadi3003@gmail.com

To Cite this Article

K S Lokesh, Dunna Nikitha Rao, C Likhitha, "Smart Intrusion Detection in Wireless Sensor Networks Using Optimized Ensemble Learning", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 04, April 2026,pp: 256-266, DOI: http://doi.org/10.64771/jsetms.2026.v03.i04.pp256-266

Submitted: 28-02-2026 Accepted: 01-04-2026 Published: 08-04-2026

Abstract: The use of WSNs is gaining significance in the real-time data gathering, environmental monitoring, and security. But it is easy to be hacked into them and undermine their security and reliability. An explainable ensemble-based intrusion detection system was developed using the Kaggle Wireless Sensor Network Dataset that was capable of addressing this issue. In the case of hyperparameter optimization, the algorithm is Particle Swarm Optimization (PSO) and GridSearchCV. It also applies various models of machine learning, including DT + PSO, RF + PSO,KNN + PSO, XGB + PSO, and a hybrid ensemble based on LightGBM and ExtraTree, (RF + DT) + PSO, (RF + KNN) + PSO, (RF + KNN + XGB) + The model results were elucidated with the help of AI methods that could be described, such as LIME and SHAP, which simplified the intrusion detection process. The experiment indicated that the stacking ensemble performed better than any single model and accurately detected threats with 98.1 precision and recall and F1-score. It demonstrates that it is a good and comprehensible solution to WSN intrusion detection. The web application is a Flask-based application that allows users to log in and add features dynamically, prepare attacks and categorize them into the list of Normals, Blackhole, Flooding, Grayhole, and TDMA.

"Index Terms: *Wireless Sensor Networks, Intrusion Detection, Particle Swarm Optimization, Ensemble Learning, Decision Tree, Random Forest, K-Nearest Neighbors, Explainable AI".*

1. INTRODUCTION

Nowadays, WSNs constitute a significant component of the modern wireless communication systems. They enable the real-time tracking and data gathering in large scale in most sectors, including healthcare, military operations, automation in industries, and monitoring the environment [1]. WSNs consist of distributed sensor nodes which perform tasks such as sensing, processing and communication. They work with limited resources and make sure that data is sent efficiently across changing network topologies [2]. They are ideal in applications that require them to execute themselves with minimal or no assistance of an individual due to their flexibility, cost-effectiveness, and scalability. Although WSNs possess the following advantages, they are open and distributed and this implies that they are susceptible to numerous threats. They include data privacy, integrity, and availability attacks and unauthorized access that harm the data [3]. Therefore, it is important to ensure that the security of WSNs is high to ensure that contact operation continues and the network does not crash.

Nevertheless, attaining good breach detection in the WSNs is not easy since the systems have certain inherent constraints, such as limited computing capabilities, memory and energy [4]. Conventional Intrusion Detection Systems (IDS) can scarcely be adapted to the special requirements of WSNs since they are based on fixed models which cannot be adjusted to the nature of the network variation over time and the variation of attack patterns [5]. Also, intrusion logs are highly skewed, with a big number of normal traffic than attacks. This complicates the detection of real threats and increases the false positive rates [6]. The accuracy of detection has trade-offs with the work required to do it, and the size of the detection that can be done with current detection techniques, whether of an anomaly-based type or a signature-based type [7]. Common methods of securing against live threats are not

sufficient as WSNs continue to grow in size and complexity [8]. The reason is that they fail to keep pace with economy of energy and reliability of communication. These issues lead us to realize that we should have smarter, more adaptable and more comprehensible intruder detection systems that can achieve an optimal balance between accuracy, openness and resource efficiency.

The ultimate aim of the study is to develop a smart intrusion detection system that is more compatible with WSNs and simpler to comprehend, modify, and deal with. The approach employs more sophisticated tools of data balancing and interpretability to ensure a better classification and create confidence in the accuracy of the predictions of the system [9]. The proposed framework also facilitates openness because it allows users and network managers to know and verify the rationale of intrusion alerts. This model is designed to enhance detection and maintain high levels of operation efficiency in low-resource WSN environments. It achieves this through the critical issues of data imbalance, the necessity to respond to evolving threats, and the necessity of explainable decisions.

This contribution is important as it can help to make the WSN infrastructures deployed in key locations more resilient and more reliable. By giving an intelligent and understandable way to identify problems, it improves situational awareness and lets people respond quickly to security events, which lowers the effect that cyber threats have on network operations [10]. Explainable intelligence added also enhances user confidence and simplifies its use in systems in the real world. This development assists in developing secure, trustworthy and energy effective WSN conditions and thus it can be said that they can be incorporated in future wireless communication systems in a manner that is both sustainable and reliable.

2. LITERATURE REVIEW

Putrada et al. [11] in their research investigated the way machine learning would enhance the performance of Intrusion IDS used in WSNs. Particularly, they analyzed the way of how to resolve the data mismatch that is a widespread issue in network security analytics. It is the XGBoost algorithm that assisted them to make right calls in labeling attacks even in cases when datasets were not balanced and underline how accurate and scalable it is. Their tests revealed that the XGBoost was more effective compared to the other classifiers in the context of recognition rate and accuracy. Nevertheless, the technique continued to struggle with infrequent types of attacks and remain steady when conditions in the network varied. Dharini et al. [12] researched the detection of intrusions in WSNs by developing a model, which involved boosting based on machine learning on the LEACH Denial of Service (DoS) attack data. In their experiment, they examined a variety of various techniques to boosting and discovered that ensemble-based models performed significantly better in terms of correctly classifying attack patterns in grouped network environments. The authors emphasized the significance of adaptive learning in terms of the WSN security, particularly its role in avoiding energy wastage and DoS attacks. But, they also noted that real-time intrusion reaction was still not being achieved because of the high cost of computing.

Tan et al. [13] developed a method through which the WSNs can detect intrusions and it relies on the Random Forest algorithm and the SMOTE. They resolved the issue of mismatch in the dataset by generating the artificial samples of the minority group, which simplified the classification of the uncommon types of attacks. The study revealed that random forest was superior at locating things as compared to the conventional decision-tree-based classifiers and was also effective even in case of noise in the data. Although it was effective in equalizing datasets and ensuring the system was more reliable, the model was discovered to require improvement in its aspect of the cost incurred to operate it as well as its ability to adapt to new attacks. In the study by Abhale and Reddy [14], deep learning was applied to enhance intrusion detection in the WSNs by applying neural architectures capable of identifying complex, non-linear patterns in the network data. Their study revealed that DLmodels, particularly the CNNs and LSTM networks had a higher level of detecting complex threats compared to shallow classifiers. Nevertheless, the authors admitted that the models might be difficult to comprehend and required to be implemented in a manner that consumed less energy in the WSN nodes that have limited resources.

According to Hemanand et al. [15], a smart system to find and sort intrusions, which employs a CSGO-LSVM model was proposed. The hybrid approach enhanced the correctness of the classification as well as the simplicity of the calculations as the most suitable features and model parameters were selected. They were successful in telling the difference between various types of attacks in WSNs and this indicates that their approach could be generalized to a large number of datasets. The research did indicate however that scale and real-time flexibility may be inhibited by the complexity of the models and the use of parameter tuning. Similarly, Chandre et al. [16] demonstrated the process of utilizing CNNs to develop an intrusion detection system to WSNs. They were

designed in such a way that they were aimed at proactive protection in that unusual activity was identified before it could interfere with the network integrity. The CNN-based model was highly precise and responded rather swiftly by extracting characteristics of patterns within the network traffic that occurred across space and time. The writers however noted that CNNs required much processing power which was an issue to the low-powered sensor devices which are prevalent in WSNs.

The authors of the article by Singh et al. [17] tested the effectiveness of a fuzzy logic-driven approach to intrusion prevention in WSNs with the help of the WSN-DS dataset. They had a system that employed fuzzy rules of inference in addressing doubt in network behavior. This was because it was now possible to detect small changes that might translate to an intrusion. The fuzzy approach was good at minimizing false positives as well as ensuring that learning was able to adjust to the evolving situations in the environment. The authors were however aware that finetuning of fuzzy parameters to achieve optimal results may require a lot of time and experience. In an effort to assist WSNs to identify more complex cyberattack, Gowdhaman and Dhanapal [18] developed an IDS using a deep neural network. When trained on large data sets, their deep learning model achieved a lot in terms of identifying various forms of attacks such as spoofing, blackhole and Sybil attacks. The findings indicated that deep neural designs would achieve high detection rates at reduced false alarms rates. The authors said however that the model training exercise was strenuous on computers and required further effort to ensure that it would operate with the constraints of WSN nodes.

Behiry and Aly [19] explained a hybrid type of intrusion detection technique, which utilizes the AI and ML approaches, with a feature reduction procedure that increases the speed of the system. They have employed dimensionality reduction algorithms to select the most appropriate sets of features and this made it more accurate and quicker to process. The hybrid model in large WSNs with a significant number of various types of data found it easy to discover cyberattacks. The authors emphasized that intelligent choice of features did not only simplify models, but also increased the lifespan of sensor nodes by reducing the volume of their workload. However, the capability of the model to adapt to new patterns of attacks that had never been witnessed before required further studies. Gebremariam et al. [20] contributed to the research by developing an intrusion detection system of hierarchical WSNs based on hybrid machine learning. They tried to select a good balance between accuracy and computational efficiency among various network layers by applying both supervised and unsupervised learning models. The proposed approach could identify a variety of attacks and at the same time be energy efficient which is highly significant in WSN environments. Although the model was effective, it could not be scaled and understood easily particularly when it comes to other forms of network topologies.

3. MATERIALS AND METHODS

With the AWID used to locate odd network action, the proposed system provides an elucidable Intrusion Detection Framework to [21] WSNs. It is a combination of advanced ML techniques enhanced by PSO and GS when tuning hyperparameters to perform a more favorable classification and make it more adaptable. Base and ensemble models are used in the system. They are (DT + PSO), (RF + PSO), (KNN + PSO), (XGB + PSO) and a combination of them such as (RF + DT + KNN) + PSO. A stacking classifier which uses LightGBM and ExtraTree with explainable AI procedures LIME and SHAP also provides interpretability. To have real time and transparent intrusion detection, the system is configured using a Flask based web service.

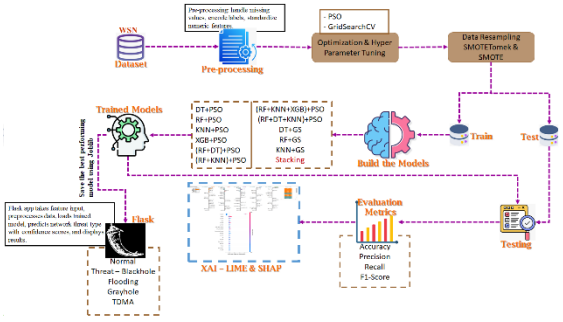

Fig.1 Proposed Architecture

Figure 1 illustrates a machine learning procedure of detecting threats in WSNs. It includes pre-processing, optimization through PSO/GridSearchCV as well as SMOTE resampling. Various models are instructed and experimented, and then scored with standard measures and XAI (LIME/SHAP), and lastly they are deployed to an application with a Flask application.

a) Dataset Collection:

Wireless Sensor Networks WSN-DS dataset contains 374,661 records with 19 attributes that indicate what sensor nodes are performing in [25]. It contains such information as the node ID, the simulation time, the state of the cluster head, the number of communications (ADV, JOIN, SCH, DATA), the distance between nodes, the amount of energy consumed, and labels that indicate the normality or an attack behavior. This is a dataset based on simulated WSN environments that contains temporal, spatial and communication-based data that are required in the intrusion detection. It also allows you to check network performance, energy saving and security behaviour under a broad spectrum of operational situations.

	id	Time	Is_CH	who CH	Dist_To_CH	ADV_S	ADV_R	JOIN_S	JOIN_R	SCH_S	SCH_R	Rank	DATA_S
0	101000	50	1	101000	0.00000	1	0	0	25	1	0	0	0
1	101001	50	0	101044	75.32345	0	4	1	0	0	1	2	38
2	101002	50	0	101010	46.95453	0	4	1	0	0	1	19	41
3	101003	50	0	101044	64.85231	0	4	1	0	0	1	16	38
4	101004	50	0	101010	4.83341	0	4	1	0	0	1	25	41

Fig.2 WSN Dataset

b) Pre-Processing:

The preprocessing pipeline of the WSN-DS dataset consists of exploratory analysis, data cleaning, feature scaling, balancing of classes and train-test splitting. This ensures that the information is of desirable quality, balanced and well prepared to be used in the detection of intrusions.

Exploratory Data Analysis (EDA): We start by analyzing the WSN-DS dataset in the first step to gain knowledge about the way it is organized, the types of data, and the distribution of classes in it. In order to perform basic EDA, you summarize feature characteristics and data balance with the help of some summary functions. Bar charts will allow you to see the distribution of the normal and attack classes and this may allow you to identify the gaps and inform the preprocessing and balancing process to make the model more trustworthy.

Data Preprocessing: At this point, the information is changed to make sure it is of good quality and consistency. To make the things consistent, additional columns are removed, the name of the categories is coded with the help of LabelEncoder, and the numbers are normalized with the help of the StandardScaler. The operations prepare the dataset to undergo machine learning by removing noise, scaling the features in a better way, and ensuring that all attributes contribute equally to the training process to achieve effective Wireless Sensor Networks intrusion detection.

Data Balancing: The issues of class mismatch in the dataset are addressed by means of such data balancing techniques as RUS and SMOTETomek. These methods ensure that the classes are equally represented hence no favoritism against classes that have those in the majority. Resampling the data increases the model making it more general and stable and thus higher in telling the difference between normal and attack cases in intrusion detection.

c) Training and Testing:

The balanced and preprocessed data is split into two groups training and testing at 80:20. The training data is to produce and improve machine learning models and the testing data is to check the functioning and the possibility of their use in other cases. StandardScaler makes sure that the scale is the same for both sets. This division allows objective model validation to occur, which allows proper assessment of detection abilities in the detection of malicious practices in WSN.

d) Algorithms:

DT + PSO: uses features to learn decision rules and classify data of a wireless sensor network. Higher accuracy and reduced overfitting are then optimized by PSO. This [24] simplifies and is easy to manipulate and identify malicious activities with large volumes of data to ensure proper intrusion detection.

$$I\,(i) = 1 - \sum_{i=1}^{k} p_i^2 \quad (1)$$

RF + PSO: optimizes the combination of decision trees to explore attacks more efficiently and PSO optimizes the parameters of the trees to make them more stable and accurate. [26] Enhances reliability, reduces the number of false reports, and ensures that the security of Wireless Sensor Network could be tracked precisely and in large volumes.

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \quad (2)$$

KNN + PSO: Determines unusual behavior in the data of Wireless Sensor Networks utilizing neighbor-based classification and PSO utilizing the most of the neighbor count and distance measurements. [27] It is more accurate, can handle various data types, and is able to identify insidious intrusions.

$$distance(x, X_i) = \sqrt{\sum_{j=1}^{d} \left(x_j - X_{i_j}\right)^2} \quad (3)$$

XGB + PSO: Sequential trees are used to learn complex threats on networks and PSO is used to optimize the parameters of boosting in order to be more balanced and accurate. Enhances the resilience to skewed data, reduces overfitting, and increases the strength of intrusion detection in dynamic environments [28].

RF + DT + PSO: uses ensemble and interpretable decision models, which have been optimized by PSO to appropriately classify information on a wireless sensor network. [29] Strikes a balance between accuracy and explainability, thus it becomes easier to detect attacks and reduces the number of false alarms in order to provide reliable security analysis.

RF + KNN + PSO: integrates group learning and learning at a distance with the assistance of PSO in order to locate network features based attacks. assists in classifying atypical attacks in a more effective way, makes systems less sensitive to noise data and ensures the ability of the Wireless Sensor Networks to be monitored correctly and in large scale.

RF + KNN + XGB + PSO: It identifies both local and global patterns of intrusion using PSO optimization and ensemble, distance and boosting techniques. Improves the detection accuracy, reduces false positives and is applicable in Wireless Sensor Networks that experience varying attack patterns.

RF + DT + KNN + PSO: optimizes ensemble, interpretable, and instance-based models using PSO to discover a large set of threats. [30] Aims at achieving a compromise between accuracy, scaling, and readability, it provides decent intrusion detection in a Wireless Sensor Network with varying conditions.

DT + GS: Optimizes decision tree hyperparameters by using GridSearchCV. Enhances the readability, accuracy and reliability of detecting malicious activities in Wireless Sensor Network environment.

RF + GS: Tunes group decision trees with the help of the Uses GridSearchCV to obtain the best performance parameters. Its ability to process complex and high-dimensional data on Wireless Sensor Networks with accuracy, stability, and effectiveness is good.

KNN + GS: The best results of neighbor count and distance measures would be obtained with the help of gridSearchCV when it comes to neighbor-based classification. It is more precise in terms of detection, it is able to find rare intrusions with accuracy and it operates well in dynamic conditions of Wireless Sensor Network.

Stacking with LightGBM and ExtraTree: Gradient boosting and tree-based ensemble models are put together in a meta-learning structure. Integrates predictions to enhance their accuracy, reliability and generalizability, ensuring the intrusion detection is dependable and adaptable to most situations in a wide variety of Wireless Sensor Networks.

$$y = g(Y_{base}) = g\left(f_1(x), f_2(x), \ldots., f_m(x)\right) \quad (4)$$

e) Integration of XAI and Flask Framework:

The system employs XAI methods, particularly, LIME and SHAP, to demystify and simplify the predictions of the intrusion detection model of Wireless Sensor Networks. LIME works out local explanations by modifying some test cases and testing the response of the model, to demonstrate how each feature influenced the projected class. This will make you know how the model makes decisions of what to do in some network scenarios such as Grayhole or Blackhole attacks. Besides LIME, SHAP offers global interpretability based on the measure of the impact of each feature on multiple cases. This allows you to view the features of importance and how they influence other examples, hence the recognition system is more reliable.

The Flask web application is employed in the installation of the framework and this facilitates an easy application in real life since it offers an interactive platform where real-time intrusion is detected. A simple interface allows one to enter network information, view estimates, and receive XAI explanations. This unification combines machine learning and real-life applications, a blend of high precision in detection and user-friendliness and comprehensibility. The integration of XAI and Flask ensures openness, responsibility, and flexibility of the system and it is therefore possible to monitor and manage WSN security in the ever changing network environments.

4. EXPERIMENTAL RESULTS

Accuracy: The ability of a test to distinguish between unhealthy and healthy individuals is referred to as its accuracy. In order to get a clue of the degree to which a test is correct we ought to find out the percentage of cases that are true positives as well as true negatives. This can be represented as in math.

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} (5)$$

Precision: Precision Percentage of the cases or samples that have been correctly classified to those correctly classified as positives. The way to determine the precision is, then, as follows:

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{False Positive}} (6)$$

Recall: Recall is a metric used in machine learning MLto indicate the ability of a model to identify all the important instances of a particular class. It demonstrates the extent to which a model represents the instances of a specific class. It is determined by the number of correct predictions of the positive observations by the total number of the real positives.

$$Recall = \frac{\text{TP}}{\text{TP } + \text{ FN}} (7)$$

F1-Score: The F1 score is the method of quantifying the accuracy of a machine learning model. It sums up the accuracy and the recall scores of a model. The accuracy measure measures the number of times, in the entire data, a model made a correct guess.

$$F1\ Score = 2 * \frac{Recall \text{ X } Precision}{Recall + Precision} * 100 \ \ (8)$$

Table.1 Performance Evaluation Table

ML Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
DT + PSO	0.966	0.968	0.966	0.966
RF + PSO	0.976	0.977	0.976	0.976
KNN + PSO	0.960	0.962	0.960	0.960
XGB + PSO	0.971	0.972	0.971	0.971
(RF + DT) + PSO	0.973	0.975	0.973	0.973
(RF + KNN) + PSO	0.971	0.973	0.971	0.971
(RF + KNN + XGB) + PSO	0.976	0.977	0.976	0.976
(RF + DT + KNN) + PSO	0.975	0.976	0.975	0.975
DT + GS	0.968	0.969	0.968	0.968
RF + GS	0.979	0.979	0.979	0.979
KNN + GS	0.957	0.958	0.957	0.957
Extension Stacking	**0.981**	**0.981**	**0.981**	**0.981**

The table 1 compares the ML models in terms of the accuracy, precision, recall and F1-score. It demonstrates that Stacking model performs the best, scoring the highest points on all the performance metrics.

Fig.3 Comparison Graph

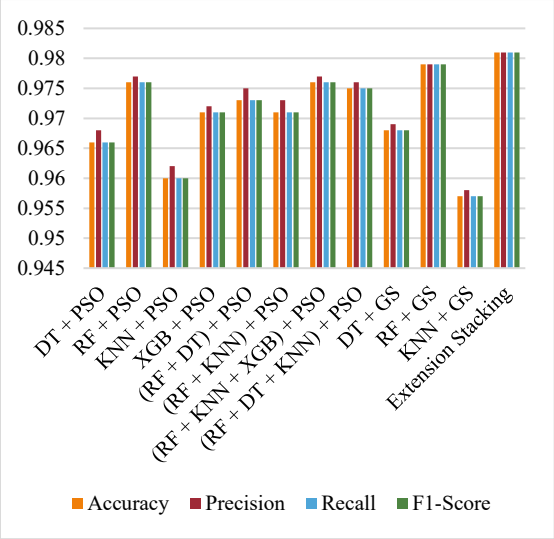

Figure 3 indicates the accuracy, the precision, the recall, and the F1-score of each model in the ML, where the various colors depict the performance of the models. The most performing model is stacking.

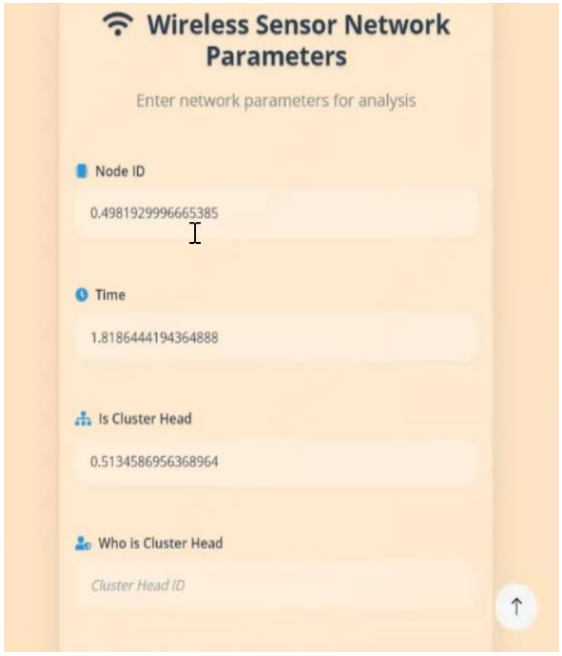

Fig.4 Enter Input Data

Fig. 4 depicts the interface of the user input of a Wireless Sensor Network data to obtain real-time intrusion detection.

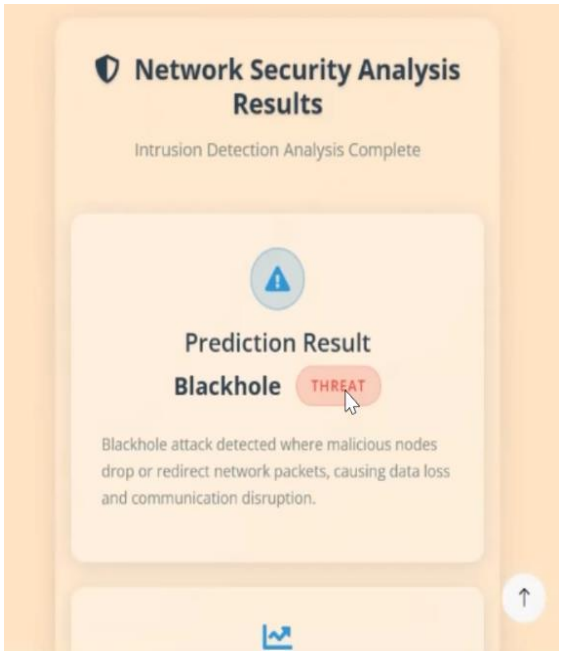

Fig.5 Predicted Result

The expected result in Fig. 5 is "Black hole," which means that a specific intrusion into the network has been found.

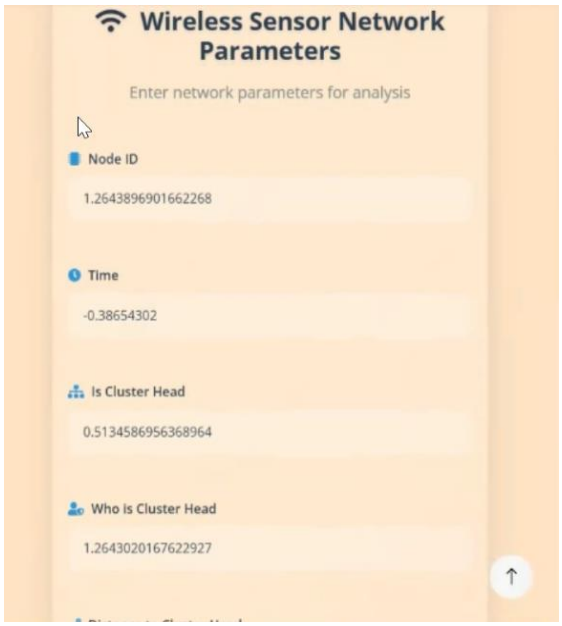

Fig.6 Enter Input Data

Fig. 6 has an input screen shot of a form that allows users to provide settings of a Wireless Sensor Network to forecast and classify network intrusions.

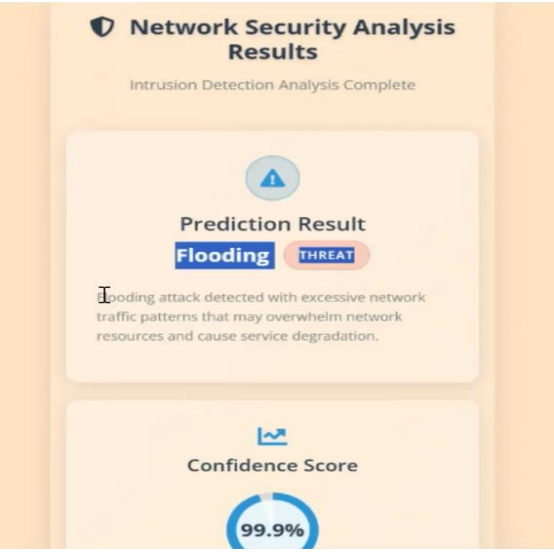

Fig.7 Predicted Results

Fig. 7 is expected to display the outcome of the expected result as Flooding, indicating that the appropriate type of network attack was detected successfully.

5. CONCLUSION

The new explainable ensemble-based intrusion detection system of the WSNs was quite effective in locating bad behavior and was dependable and simple to interpret. The integration of Particle Swarm Optimization with other different machine learning models and ensemble techniques enhanced feature selection, enhanced reliability of the system and reduced the false alarms in cases where the network is evolving. The methods of explainable AI, such as LIME and SHAP, enhanced transparency by showing the characteristics that were considered in each prediction. This is highly significant in the case of WSN environments which require to be secure. It was experimentally demonstrated that the stacking ensemble classifier consisting of LightGBM and ExtraTree did well more often than individual models and in comparison with the hybrid models. The stacking technique was detected with an accuracy of 98.1% and a balanced precision, recall, and F1-score indicating that it was effective at detecting complicated entry patterns. In order to implement the system in the real world, the Flask framework was employed to launch the system and offer a web-based interface to the system. The rollout includes secure user sign up/sign in, real time network feature input, automated preprocessing and real time visualization of prediction. The app classifies data into categories, which include Normal, Blackhole, Flooding, Grayhole and TDMA. In general, the framework enhances the safety of WSNs and offers scalable, interpretable and user friendly intrusion detection applicable to existing sensor network implementations.

In order to enhance the precision of intrusion detection in WSNs, further research can be conducted on the integration of deep learning frameworks such as LSTM and GRU and CNN-based frameworks. In order to test real-time intrusion detection, the system can be applied to streaming environments which are capable of dealing with time varying attack patterns. The expandability of the framework can also be experimented on larger, more diverse WSN data to determine its suitability in diverse contexts. To improve the hyperparameter optimization, hybrid optimization algorithms that are a combination of PSO and more sophisticated metaheuristics such as Genetic Algorithms or Grey Wolf Optimizer may be implemented. Moreover, federated learning and edge computing can also potentially increase data privacy and reduce latency, and it could be safe and effective that distributed WSN infrastructures can detect intrusion.

REFERENCES

[1] Saleh, H. M., Marouane, H., & Fakhfakh, A. (2024). Improves Intrusion Detection Performance InWireless Sensor Networks Through Machine Learning, Enhanced By An Accelerated Deep Learning Model With Advanced Feature Selection. Iraqi Journal for Computer Science and Mathematics, 5(3), 23.

[2] Sadia, H., Farhan, S., Haq, Y. U., Sana, R., Mahmood, T., Bahaj, S. A. O., & Khan, A. R. (2024). Intrusion detection system for wireless sensor networks: A machine learning based approach. IEEE Access, 12, 52565-52582.

[3] Mopuru, B., & Pachipala, Y. (2024). Advancing IoT Security: Integrative Machine Learning Models for Enhanced Intrusion Detection in Wireless Sensor Networks. Engineering, Technology & Applied Science Research, 14(4), 14840-14847.
[4] Pandey, V. K., Prakash, S., Gupta, T. K., Sinha, P., Yang, T., Rathore, R. S., ... & Bakhsh, S. T. (2025). Enhancing intrusion detection in wireless sensor networks using a Tabu search based optimized random forest. Scientific Reports, 15(1), 18634.
[5] Singh, A., Amutha, J., Nagar, J., Sharma, S., & Lee, C. C. (2022). AutoML-ID: Automated machine learning model for intrusion detection using wireless sensor network. Scientific Reports, 12(1), 9074.
[6] M. Dener, S. Al, and A. Orman, ''STLGBM-DDS: An efficient data balanced DoS detection system for wireless sensor networks on big data environment,'' IEEE Access, vol. 10, pp. 92931–92945, 2022.
[7] S.Jiang, J. Zhao, and X. Xu, ''SLGBM: An intrusion detection mechanism for wireless sensor networks in smart environments,'' IEEE Access, vol. 8, pp. 169548–169558, 2020.
[8] G, Viswanath., N, Madhvik., K, Bhaskar., K, Supriya. (2024). Machine-Learning-Based Cloud Intrusion Detection. International Journal of Mechanical Engineering Research and Technology, 16(9), 38-52.
[9] S. Sharmin, I. Ahmedy, and R. M. Noor, ''An energy-efficient data aggregation clustering algorithm for wireless sensor networks using hybrid PSO,'' Energies, vol. 16, no. 5, p. 2487, Mar. 2023.
[10] N. M. Alruhaily and D. M. Ibrahim, ''A multi-layer machine learning-based intrusion detection system for wireless sensor networks,'' Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 4, pp. 281–288, 2021.
[11] A. G. Putrada, N. Alamsyah, S. F. Pane, and M. N. Fauzan, ''XGBoost for IDS on WSN cyber attacks with imbalanced data,'' in Proc. Int. Symp. Electron. Smart Devices (ISESD), Nov. 2022, pp. 1–7.
[12] N. Dharini, J. Katiravan, D. M. S. Priya, and S. V. A. Sakthi, ''Intrusion detection in novel WSN-leach dos attack dataset using machine learning based boosting algorithms,'' Proc. Comput. Sci., vol. 230, pp. 90–99, Jan. 2023.
[13] X. Tan, S. Su, Z. Huang, X. Guo, Z. Zuo, X. Sun, and L. Li, ''Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm,'' Sensors, vol. 19, no. 1, p. 203, Jan. 2019.
[14] A. B. Abhale and A. J. Reddy, ''Deep learning perspectives to detecting intrusions in wireless sensor networks,'' Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 2, pp. 18–26, Jan. 2023.
[15] D. Hemanand, G. Reddy, S. S. Babu, K. R. Balmuri, T. Chitra, and S. Gopalakrishnan, ''An intelligent intrusion detection and classification system using CSGO-LSVM model for wireless sensor networks(WSNs),'' Int. J. Intell. Syst. Appl. Eng., vol. 10, no. 3, pp. 285–293, Oct. 2022.
[16] P. R. Chandre, P. Mahalle, and G. Shinde, ''Intrusion prevention system using convolutional neural network for wireless sensor network,'' IAES Int. J. Artif. Intell. (IJ-AI), vol. 11, no. 2, p. 504, Jun. 2022.
[17] Lakshmi, J. M., Prasad, K. K., & Viswanath, G. (2025). Proactive Security in Multi-Cloud Environments: A Blockchain Integrated Real-Time Anomaly Detection and Mitigation Framework. Cuestiones De Fisioterapia, 54(2), 392-417.
[18] V. Gowdhaman and R. Dhanapal, ''An intrusion detection system for wireless sensor networks using deep neural network,'' Soft Comput., vol. 26, no. 23, pp. 13059–13067, Dec. 2022.
[19] M. H. Behiry and M. Aly, ''Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods,'' J. Big Data, vol. 11, no. 1, p. 16, Jan. 2024.
[20] G. G. Gebremariam, J. Panda, and S. Indu, ''Design of advanced intrusion detection systems based on hybrid machine learning techniques in hierarchically wireless sensor networks,'' Connection Sci., vol. 35, no. 1, Dec. 2023, Art. no. 2246703.
[21] M. A. Elsadig, ''Detection of denial-of-service attack in wireless sensor networks: A lightweight machine learning approach,'' IEEE Access, vol. 11, pp. 83537–83552, 2023.
[22] Naresh, M., Gudditti, M., Viswanath, & SunilKumarReddy, M. T. (2014). Distributed Utility-Based Energy Efficient Cooperative Medium Access Control in MANETS.
[23] G. Liu, H. Zhao, F. Fan, G. Liu, Q. Xu, and S. Nazir, ''An enhanced intrusion detection model based on improved kNN in WSNs,'' Sensors, vol. 22, no. 4, p. 1407, Feb. 2022.
[24] M. Aljebreen, M. A. Alohali, M. K. Saeed, H. Mohsen, M. Al Duhayyim, A. A. Abdelmageed, S. Drar, and S. Abdelbagi, ''Binary chimp optimiza tion algorithm with ML based intrusion detection for secure IoT-assisted wireless sensor networks,'' Sensors, vol. 23, no. 8, p. 4073, Apr. 2023.

[25] T. M. Nguyen, H. H.-P. Vo, and M. Yoo, ‘‘Enhancing intrusion detection in wireless sensor networks using a GSWO-CatBoost approach,’’ Sensors, vol. 24, no. 11, p. 3339, May 2024.
[26] Kumar, K., Udaya Suriya Rajkumar, D., Viswanath, G., & Mahalakshmi, J. (2024). A Hybrid Particle Swarm Optimization and C4.5 for Network Intrusion Detection and Prevention System. International Journal of Computing, 23(1), 109-115. https://doi.org/10.47839/ijc.23.1.3442
[27] N. Marriwala and P. Rathee, ‘‘An approach to increase the wireless sensor network lifetime,’’ in Proc. World Congr. Inf. Commun. Technol., Oct. 2012, pp. 495–499. [28] D. Shah, Z. Yu Xue, and T. M. Aamodt, ‘‘Label encoding for regression networks,’’ 2022, arXiv:2212.01927.
[29] F. Aldi, F. Hadi, N. A. Rahmi, and S. Defit, ‘‘Standardscaler’s potential in enhancing breast cancer accuracy using machine learning,’’ J. Appl. Eng. Technological Sci. (JAETS), vol. 5, no. 1, pp. 401–413, Dec. 2023.
[30] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, ‘‘SMOTE for learning from imbalanced data: Progress and challenges, marking the 15 year anniversary,’’ J. Artif. Intell. Res., vol. 61, pp. 863–905, Apr. 2018.