JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

SMART INTRUSION DETECTION MODEL FOR WSN: A MACHINE LEARNING APPROACH

[1]Uthra T K
UG, Student, Department of ECE
Meenakshi Sundararajan Engineering College, Chennai, Tamil Nadu, India.

[2]Sidharth M V
UG, Student, Department of ECE
Meenakshi Sundararajan Engineering College, Chennai, Tamil Nadu, India.

[3]Dr.A.Babiyola
Professor, Department of ECE
Meenakshi Sundararajan Engineering College, Chennai, Tamil Nadu, India.

Abstract : Wireless sensor networks have garnered significant adoption as a technology for collecting and analyzing data from the environment. Their versatility and wide-ranging applications, encompassing critical military operations, forest fire detection, healthcare systems, and civilian uses, have made them a focal point of extensive research. The potential of WSNs to provide valuable insights and address various real-world challenges has spurred significant interest and exploration in the scientific community. Nevertheless, Wireless Sensor Networks (WSNs) are susceptible to security threats, especially in unattended environments. To address these threats, the popular Low-Energy Adaptive Clustering Hierarchy protocol is deployed with machine learning techniques. This paper makes an analysis of different machine learning algorithms with multiple class of parameters and evaluates it with criterion as such favorable for a WSN. The best fit algorithm for intrusion detection is the Random Forest with 8 extracted features for classifying attacks such as blackhole attack, grayhole attack, TDMA and flooding attack. The accuracy score is 98.39% with prediction accuracy for attacks: 84%, 85%, 87% and 86% respectively. Unlike detecting the anomaly alone, this mechanism when given to a centralized management system become a smart governance model. This prevents the network from different intrusion methodologies and enhance the characteristics such as energy efficiency, network lifetime, throughput, performance and security issue of the wireless sensor network.

Keywords - Intrusion detection, Wireless Sensor Network, LEACH protocol, Machine learning, Smart governance.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have gained significant importance in research due to their extensive range of real-time applications. The design of WSNs is influenced by crucial factors such as scalability, fault-tolerance, and power consumption. WSNs consist of numerous autonomous sensor nodes that are strategically distributed across various areas of interest. These nodes collect important data and collaboratively transmit it wirelessly to a central node known as the sink node or Base Station. The transmission and management of data within the network heavily rely on specialized WSN protocols. One commonly used architecture in WSNs is the clustered architecture, which is facilitated by the LEACH (Low Energy Adaptive Clustering Hierarchy) protocol at the network layer. LEACH is a hierarchical clustering protocol that assumes each node possesses a radio with sufficient power to reach the base station or the nearest cluster head. However, maintaining full power transmission constantly results in energy wastage. In LEACH, non-cluster head nodes communicate in a Time Division Multiple Access fashion, while clusters employ Code Division Multiple Access techniques to minimize interference between them. Despite their numerous advantages, WSNs are susceptible to various security attacks due to their open and distributed nature, as well as the limited resources of sensor nodes. Safeguarding WSNs from these attacks poses significant challenges due to resource constraints and the vulnerabilities associated with sensor nodes.

II. EXISTING WORK

Several research studies have been conducted to address the growing security threats in Wireless Sensor Networks (WSNs). This particular work [1] focuses on enhancing the network's energy efficiency and lifespan by employing reinforcement learning (RL) protocols. The proposed system is designed to adapt to changes in the network, such as energy efficiency and mobility, enabling it to make improved routing decisions. The research also takes into consideration the legal constraints imposed on sensor nodes and presents an energy-balancing routing model based on RL. The results indicate that the proposed protocol outperforms existing energy-efficient routing protocols like Q-learning and LARCMS in terms of energy savings and network lifespan. The effectiveness of the proposed protocol is evaluated through metrics such as end-to-end delivery and packet delivery.

[2] This work addresses the issues of security in terms of distinct attacks and their solutions provided by different authors. The different layer attacks are tabulated with the ML techniques applied as a survey with remarks. This research paper [3] introduces LEACH-APP, a novel clustering protocol that builds upon LEACH while considering the specific application requirements of the network. The main objective of LEACH-APP is to enhance the Quality of Service (QoS) management in the network. The experiments conducted in the study demonstrate that LEACH-APP effectively improves the network's throughput and reduces latency, thereby providing robust and flexible QoS management capabilities. The protocol offers promising benefits in terms of overall network performance and better accommodates the specific application needs.

This particular study [4] focuses on the utilization of the LEACH protocol, which is a widely adopted hierarchical routing protocol in Wireless Sensor Networks (WSNs). The main objective is to create a specialized dataset that captures both normal and anomalous behaviours for future testing purposes. A specific approach is devised to collect data from Network Simulator 2 (NS-2) and process it to extract 23 distinct features. Custom code segments are designed to simulate attacks and gather the necessary data, resulting in the formation of the WSN-DS dataset. Multiple Artificial Neural Network (ANN) options are explored, and the collected dataset is used to train the model for the detection and classification of various types of Denial of Service (DoS) attacks. The evaluation is conducted using the WEKA toolbox, employing both the holdout and 10-Fold Cross Validation techniques. The most promising results are obtained with the 10-Fold Cross Validation method, utilizing a single hidden layer. The classification accuracies achieved are 92.8% for Blackhole attacks, 99.4% for Flooding attacks, 92.2% for Scheduling attacks, 75.6% for Grayhole attacks, and 99.8% for the normal case (without attacks).

III. PROPOSED WORK

The proposed work collects environmental data from the network which is treated to obtain the components of an intrusion detection system, such as feature extraction and modelling algorithm. Based on the obtained best system we deploy the machine learning model. Once it is done, when the parameters chosen through feature extraction from the network are given the model detects and classifies the type of attack. This methodology will prevent intrusion instances to a larger extent and enhance the characteristics of a wireless sensor network.

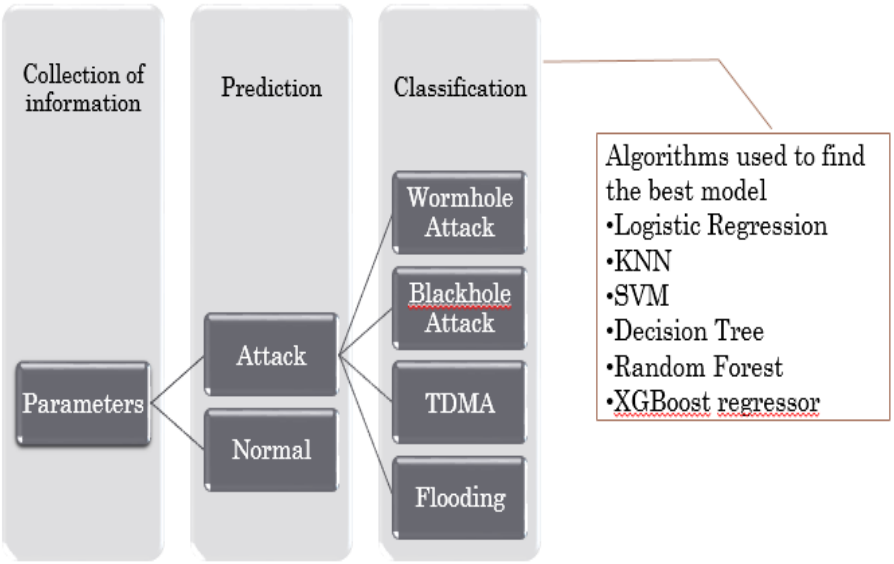

Fig. I Block diagram of the proposed solution

This model aims to pick the optimized and most accurate solution for an intrusion instance. Through progressive selection of machine learning algorithms used for classification of top 6 algorithms were chosen and used for this work.

- **Logistic Regression :** It is a supervised learning algorithm that utilizes a more sophisticated cost function known as Sigmoid or Logit function. It predicts a binary outcome either if something happens or doesn't not happen by analyzing the relationship between variables. It is efficient to train and leads to overfitting if the number of features is more than the number of observations.
- **K-Nearest Neighbors :** It is a non-parametric algorithm that relies on measuring distances to determine the category of an unknown entity. It identifies the k nearest neighbors to the target instance and assigns it to the category that appears most frequently among those neighbors. KNN can be particularly effective when the training data is large. However, determining the optimal value of k, which represents the number of neighbors to consider, can be a time-consuming process. Finding the right balance for k is crucial to ensure accurate classification results.
- **Support Vector Machine :** It is a model that represents different classes by constructing a hyperplane in a multidimensional space. SVM performs well in cases where the dataset is not excessively large, but it is known to be less effective on big datasets. However, one advantage of SVM is its robustness against outliers, as it is not highly sensitive to their presence. SVM aims to find the optimal hyperplane that maximally separates different classes, leading to effective classification performance.
- **Decision Tree :** It is a supervised learning technique that organizes data into precise classes by recursively splitting it into similar categories. It follows a flowchart-like structure, starting from the trunk and branching out to leaves, where

categories become increasingly specific. While decision trees consider all possible outcomes, they can face overfitting issues. To address this, the Random Forest algorithm combines multiple decision trees, mitigating overfitting and improving generalization.

- **Random Forest :** It is an ensemble learning method that offers a solution to complex problems. It functions as a classifier and consists of multiple decision trees trained on different subsets of the dataset. By averaging the predictions of these trees, Random Forest enhances the accuracy of predictions. It effectively addresses the overfitting problem encountered in individual decision trees. However, it should be noted that Random Forest may require additional training time due to its ensemble nature.
- **XGBOOST Regressor :** Extreme Gradient Boost is a machine learning library that offers scalable and distributed gradient-boosted decision tree algorithms. It employs a sequential approach to build decision trees, boosting their performance through parallel tree boosting. XGBoost is known for its strong performance, scalability, and interpretability, as well as its ability to handle missing values in data. However, it's important to note that XGBoost's computational complexity can be high, and in certain cases, it may be prone to overfitting.

IV. Results and Discussion

The research was conducted using Jupyter Notebook, and multiple sets of results were generated and analyzed.

4.1 Data Pre-processing

Data Pre-processing is crucial for designing a machine learning model as it is the first step to prepare the raw data fitting to the model.

	count	mean	std	min	25%	50%	75%	max
id	374661.0	274969.325879	389898.554898	101000.0	107093.00000	116071.00000	215072.00000	3.402096e+06
Time	374661.0	1064.748712	899.646164	50.0	353.00000	803.00000	1503.00000	3.600000e+03
Is_CH	374661.0	0.115766	0.319945	0.0	0.00000	0.00000	0.00000	1.000000e+00
who CH	374661.0	274980.411108	389911.221734	101000.0	107096.00000	116072.00000	215073.00000	3.402100e+06
Dist_To_CH	374661.0	22.599380	21.955794	0.0	4.73544	18.37261	33.77600	2.142746e+02
ADV_S	374661.0	0.267698	2.061148	0.0	0.00000	0.00000	0.00000	9.700000e+01
ADV_R	374661.0	6.940562	7.044319	0.0	3.00000	5.00000	7.00000	1.170000e+02
JOIN_S	374661.0	0.779905	0.414311	0.0	1.00000	1.00000	1.00000	1.000000e+00
JOIN_R	374661.0	0.737493	4.691498	0.0	0.00000	0.00000	0.00000	1.240000e+02
SCH_S	374661.0	0.288984	2.754746	0.0	0.00000	0.00000	0.00000	9.900000e+01
SCH_R	374661.0	0.747452	0.434475	0.0	0.00000	1.00000	1.00000	1.000000e+00
Rank	374661.0	9.687104	14.681901	0.0	1.00000	3.00000	13.00000	9.900000e+01
DATA_S	374661.0	44.857925	42.574464	0.0	13.00000	35.00000	62.00000	2.410000e+02
DATA_R	374661.0	73.890045	230.246335	0.0	0.00000	0.00000	0.00000	1.496000e+03
Data_Sent_To_BS	374661.0	4.569448	19.679155	0.0	0.00000	0.00000	0.00000	2.410000e+02
dist_CH_To_BS	374661.0	22.562735	50.261604	0.0	0.00000	0.00000	0.00000	2.019349e+02
send_code	374661.0	2.497957	2.407337	0.0	1.00000	2.00000	4.00000	1.500000e+01
Expaned_Energy	374661.0	0.305661	0.669462	0.0	0.05615	0.09797	0.21776	4.509394e+01

Fig. II Data description

4.2 Evaluation Criterion

The main goal of grouping the parameters is to analyze and fetch the maximal accuracy with optimal number of parameters essential for classification.

TABLE I Parameters grouping

Number Of Parameters	Parameters
4	' who CH',' Is_CH',' Dist_To_CH',' dist_CH_To_BS'
6	' who CH',' Is_CH',' Dist_To_CH',' DATA_R',' dist_CH_To_BS',' Data_Sent_To_BS'
8	' Is_CH',' who CH',' Dist_To_CH',' ADV_R',' DATA_R',' dist_CH_To_BS',' Data_Sent_To_BS',' send_code '

15	' Is_CH',' who CH',' Dist_To_CH',' ADV_S',' ADV_R',' JOIN_S',' JOIN_R',' SCH_S',' SCH_R','Rank','Rank',' DATA_S',' DATA_R',' dist_CH_To_BS',' Data_Sent_To_BS',' send_code'
16	' Is_CH',' who CH',' Dist_To_CH',' ADV_S',' ADV_R',' JOIN_S',' JOIN_R',' SCH_S',' SCH_R','Rank',' DATA_S',' DATA_R',' dist_CH_To_BS',' Data_Sent_To_BS',' send_code ','Expaned_Energy'

- ***FOUR PARAMETERS*** **-** These parameters are grouped as they give the basic structure of the network whether the node is the cluster head, the id of the cluster head, the distance between the cluster head, node and the base station in the current round.

- ***SIX PARAMETERS*** **-** These parameters are grouped such that they provide the number of data packets received from the cluster head and sent to the base station in addition to the basic information of the network in the current round given by the previous class of 4 parameters.

- ***EIGHT PARAMETERS*** **-** These parameters are grouped with the number of channel advertisement messages received from cluster head and the cluster sending code in addition to the basic network information and packet transfer data base in the current round given by the previous class of 6 parameters.

- ***FIFTEEN PARAMETERS*** **-** All the parameters obtained from the network information except the expanded amount of energy which can be precisely given as data only after performance of the network, is grouped together to check if it could produce the maximum accuracy of predicting the attack type in the current round.

- ***SIXTEEN PARAMETERS*** **-** The amount of expanded energy is taken into consideration in addition to all the parameters grouped in the class if 15 parameters and cross verify the difference in the accuracy of the prediction of attack type in the current round.

4.3 Experimental Analysis

Experiment-1: This experiment was conducted with extracted features of count 4 and the report shows Random Forest to be the best, but the number of features doesn't seem to be optimal.

TABLE II Accuracy report of experiment-1

Algorithm	Accuracy
Logistic Regression	0.9336
K-Nearest Neighbor	0.9400
Decision Tree	0.9490
Support Vector Machine	0.9478
Random Forest	0.9511
Xgboost Regressor	0.9240

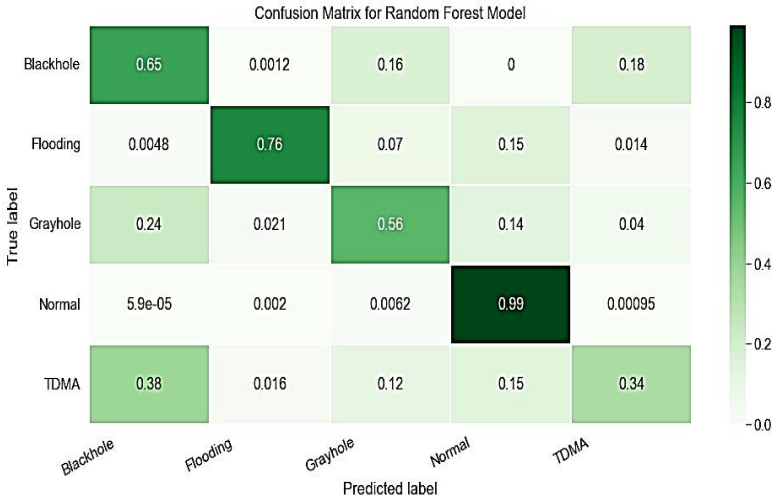

Fig. III Confusion matrix of random forest for 4 parameters

Experiment-2: This experiment was conducted with extracted features of count 6 and the report shows Random Forest to be the best, but the number of features doesn't seem to be optimal.

TABLE III Accuracy report of experiment-2

Algorithm	Accuracy
Logistic Regression	0.9313
K-Nearest Neighbor	0.9618
Decision Tree	0.9658
Support Vector Machine	0.9697
Random Forest	0.9746
Xgboost Regressor	0.9529

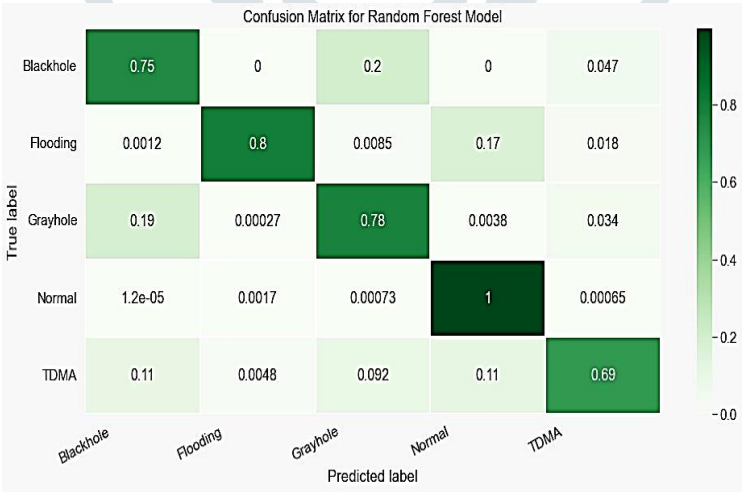

Fig. IV Confusion matrix of random forest for 6 parameters

Experiment-3: This experiment was conducted with extracted features of count 8 and the report shows Random Forest and XGBoost Regressor to be equally likely best and the number of features seem to be comparatively optimal.

TABLE IV Accuracy report of experiment-3

Algorithm	Accuracy
Logistic Regression	0.9159
K-Nearest Neighbor	0.9700
Decision Tree	0.9690
Support Vector Machine	0.9774
Random Forest	0.9839
Xgboost Regressor	0.9846

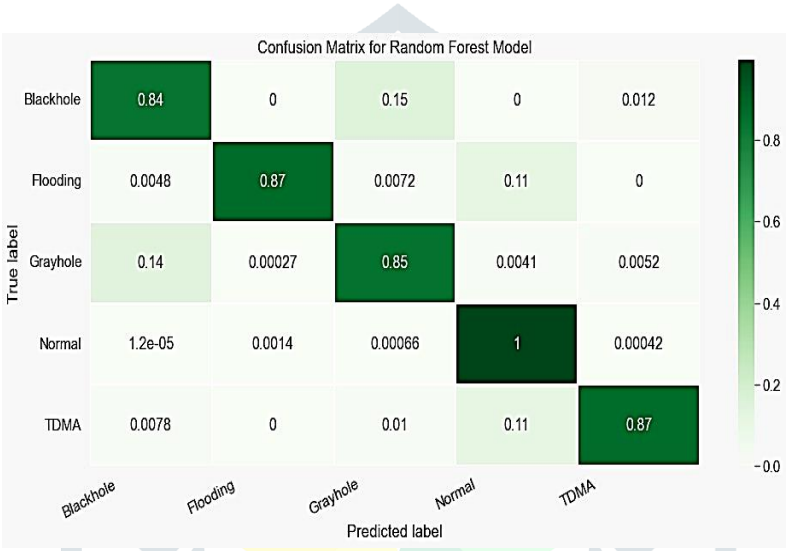

Fig. V Confusion matrix of random forest for 8 parameters

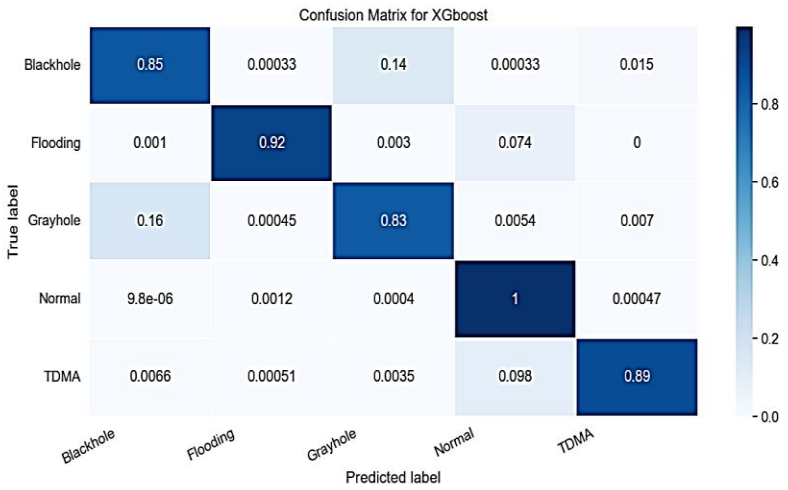

Fig. VI Confusion matrix of xgboost for 8 parameters

Experiment-4: This experiment was conducted with extracted features of count 4 and the report shows Random Forest to be the best, but the number of features doesn't seem to be optimal.

TABLE V Accuracy report of experiment-4

Algorithm	Accuracy
Logistic Regression	0.9427
K-Nearest Neighbor	0.9740
Decision Tree	0.9718
Support Vector Machine	0.9820
Random Forest	0.9869
Xgboost Regressor	0.9870

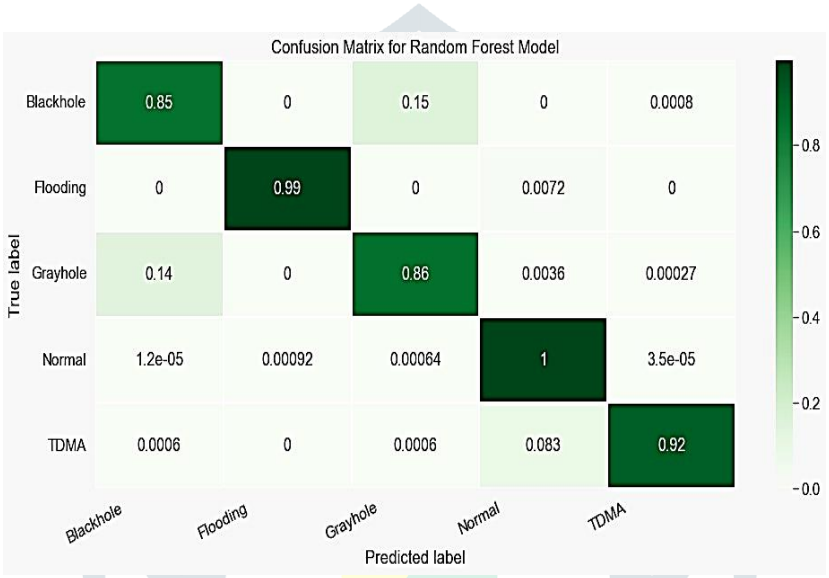

Fig. VII Confusion matrix of random forest for 15 parameters

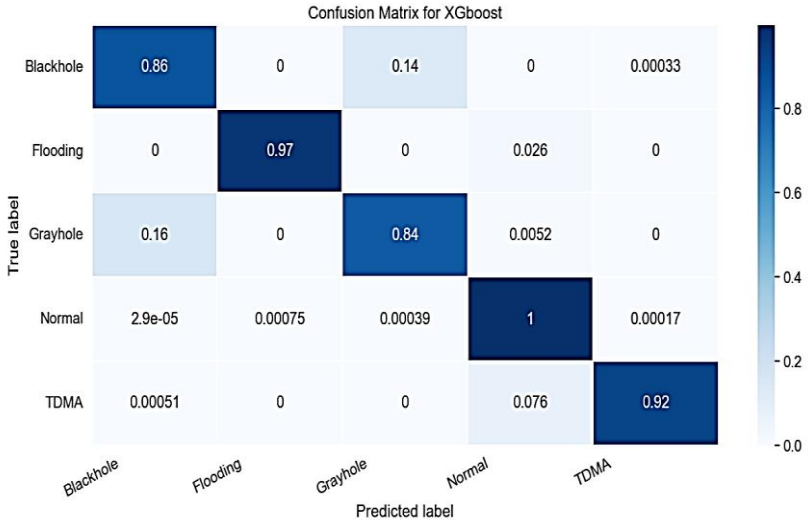

Fig. VIII Confusion matrix of xgboost for 15 parameters

Experiment-5: This experiment was conducted with extracted features of count 4 and the report shows Random Forest to be the best, but the number of features doesn't seem to be optimal.

TABLE VI Accuracy report of experiment-5

Algorithm	Accuracy
Logistic Regression	0.9572
K-Nearest Neighbor	0.9805
Decision Tree	0.9802
Support Vector Machine	0.9829
Random Forest	0.9964
Xgboost Regressor	0.9969

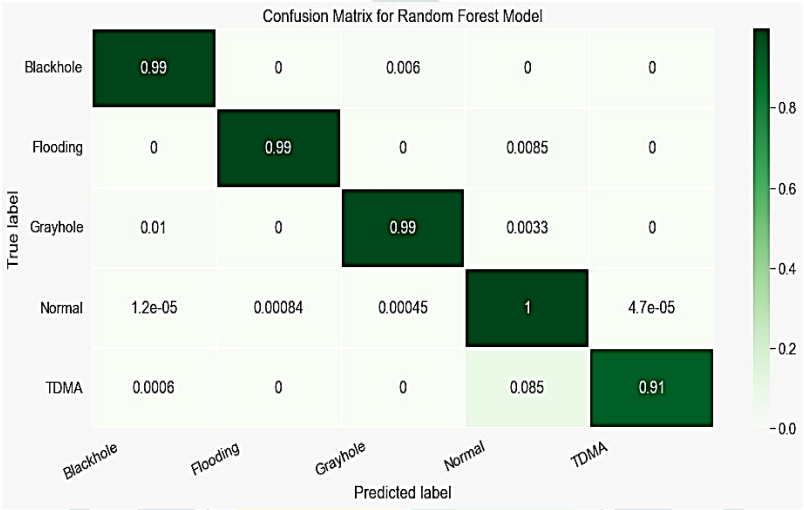

Fig. IX Confusion matrix of random forest for 16 parameters

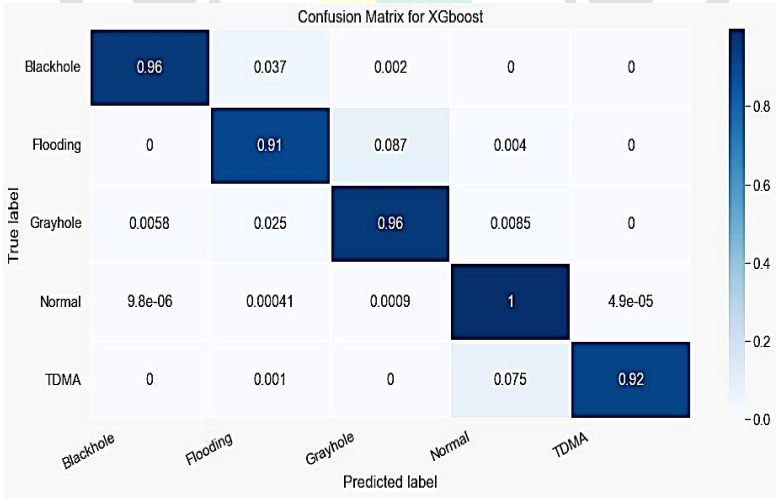

Fig. X Confusion matrix of xgboost for 16 parameters

4.3 Best-fit Algorithm

The "best fit" model is typically the one that achieves the highest accuracy or lowest error on this dataset. However, what constitutes the "best fit" can depend on various factors, such as the application and the desired trade-offs between accuracy, speed, and resource requirements. Based on the experiment analysis performed for different class of grouped parameters and considering their accuracy, precision, recall, confusion matrix we can find the 8 parameters class to be optimal. Out of the two regression models and four classification models with 8 parameters class we can conclude RANDOM FOREST to be the appropriate choice that is able to match our goal set for the intrusion detection model. From the above analysis we can observe that the accuracy scores of XGBOOST and RANDOM FOREST are almost equal. Comparing other features of the models, the training score in XGBOOST is low than that in Random Forest. Also, with respect to the power and time taken for computation being more in XGBOOST Regressor, we select "RANDOM FOREST" over XGBOOST for intrusion detection.

TABLE VII Analysis of different algorithms and their accuracy score

ALGORITHM	ACCURACY				
	4 parameters	6 parameters	8 parameters	15 parameters	16 parameters
LOGISTIC REGRESSION	0.9336	0.9313	0.9159	0.9427	0.9572
K-NEAREST NEIGHBOR	0.9400	0.9618	0.9700	0.9740	0.9805
DECISION TREE	0.9490	0.9658	0.9690	0.9718	0.9802
SUPPORT VECTOR MACHINE	0.9478	0.9697	0.9774	0.9820	0.9829
RANDOM FOREST	0.9511	0.9746	0.9839	0.9869	0.9960
XGBOOST REGRESSOR	0.9240	0.9529	0.9846	0.9870	0.9967

V. Conclusion and Future Scope

This work compares different machine learning models with different feature extracted classes and finds the best fit making it a self-sustainable governing model. Altogether it forms a centralized governing system which is also an intelligent intrusion detection system. This helps us prevent the network from different network layer attacks and enhance the characteristics such as network lifetime, throughput, performance, security issues and energy efficiency of the wireless sensor network.

This methodology proposed in this project can be scaled across various protocols in the future by changing some parameters in the code as per requirement. Also, if we can connect the device credentials used in the wireless sensor network to an interface which is connected to a cloud, it will be efficient to monitor the nodes from the interface and the suggested machine learning model can be deployed in the cloud such that it doesn't overload the centralized governing interface. The current and future versions can be released for this and installed as per the requirement of the growth of the technology.

References

[1] Simon, J. (2022). An Energy Efficient Routing Protocol based on Reinforcement Learning for WSN. IRO Journal on Sustainable Wireless Systems, 4(2), 79-89. doi:10.36548/jsws.2022.2.002.

[2] R. Kaur and J. Kaur Sandhu, "A Study on Security Attacks in Wireless Sensor Network," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2021, pp. 850-855, doi: 10.1109/ICACITE51222.2021.9404619.

[3] Alba Rozas, Alvaro Araujo, "An Application-Aware Clustering Protocol for Wireless Sensor Networks to Provide QoS Management", Journal of Sensors, vol. 2019, Article ID 8569326, 11 pages, 2019. https://doi.org/10.1155/2019/8569326.

[4] Iman Almomani, Bassam Al-Kasasbeh, Mousa AL-Akhras, "WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks", Journal of Sensors, vol. 2016, Article ID 4731953, 16 pages, 2016. https://doi.org/10.1155/2016/4731953.

[5] A. Alsadhan and N. Khan, (2013) "A proposed optimized and efficient intrusion detection system for wireless sensor network," International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, vol. 7, no. 12, pp. 1621–1624.

[6] A. Braman and G. R. Umapathi, (2014) "A comparative study on advances in LEACH Routing protocol for wireless sensor networks: a survey," International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, no. 2, pp.5883–5890.

[7] Bendigeri KY, Mallapur JD, Kumbalavati SB (2021) Direction Based Node Placement in Wireless Sensor Network. In 2021, International Conference on Artificial Intelligence and Smart Systems (ICAIS), pp 1306–1313.

[8] I. Butun, S. D. Morgera, and R. Sankar, (2014) "A survey of intrusion detection systems in wireless sensor networks," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 266–282.

[9] M. Tripathi, M. S. Gaur, and V. Laxmi, (2013) "Comparing the impact of black hole and gray hole attack on LEACH in WSN," Procedia Computer Science, vol. 19, pp. 1101–110.

[10] S. Khan and K.-K. Loo, (2009) "Real-time cross-layer design for a largescale flood detection and attack trace-back mechanism in IEEE 802.11 wireless mesh networks," Network Security, vol. 2009, no.5, pp. 9–16.

[11] Soni S, Shrivastava M (2018). Novel learning algorithms for efficient mobile sink data collection using reinforcement learning in wireless sensor network. Wireless Communications and Mobile Computing. 2018 Aug 16.

[12] S. Taneja, (2015) "An energy efficient approach using load distribution through LEACH-TLCH protocol," Journal of Network Communications and Emerging Technologies (JNCET), vol. 5, no. 3, pp. 20–23.

[13] W. -K. Yun and S. -J. Yoo, (2021) "Q-Learning-Based Data-Aggregation-Aware Energy-Efficient Routing Protocol for Wireless Sensor Networks," in IEEE Access, vol. 9, pp. 10737-10750.

[14] Wenjing Guo, Cairong Yan, and Ting Lu (2019). Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing. International Journal of Distributed Sensor Networks, 15(2):1550147719833541.

[15] Yau, KL.A., Goh, H.G., Chieng, D. et al. (2015) Application of reinforcement learning to wireless sensor networks: models and algorithms. Computing 97, 1045–1075.