Anomaly Detection System for Wireless Sensor Networks using Machine Learning

Sumit Vikram Tripathi [1], Dr. Sumana M [2]

PG Student, Deptt.of ISE, Ramaiah Institute of Technology. E-mail: sumitvt@outlook.com

Dr. Sumana M [2], Associate Professor, Deptt. of ISE, Ramaiah Institute of Technology.E-mail:sumana.m@msrit.edu

Abstract— Wireless Sensor Networks (WSNs) are turning into a more famous area of concentrate in software engineering since they have so many purposes and could prompt new ideas. This has prompted a cooperative report field where clients, specialists in application spaces, equipment producers, and programming journalists should work intently together to make frameworks that function admirably. Utilizing WSNs to do things like track down shortcomings, track down interruptions, and watch out for things is hard a direct result of tracking down irregularities. Anomaly detection techniques should be planned in light of the restrictions of sensor organizations, with the goal that sensor hubs use as little energy as could really be expected and the organization endures as far as might be feasible.

Keywords—*Wireless Sensor Networks (WSNs),Anomaly Detection System (ADS),Denial of Service (DoS) Attacks, Intrusion Detection System (IDS).*

I. INTRODUCTION

Wireless sensor networks are turning into a more famous area of concentrate in software engineering since they have so many purposes and could prompt new ideas.WSNs are dependent upon security dangers and assaults since they are fanned out and have restricted assets. Anomaly Detection Systems (ADS) are vital for finding these dangers, which makes it feasible for WSNs to work dependably and securely. In WSNs, the primary reason for an Advertisements is to track down destructive activities, unapproved access, and weird way of behaving in the network.The recommended model for finding abnormalities in WSNs utilizes both ML and deep learning models.The model displayed in Figure 1.1 is comprised of five fundamental parts. Informational index assortment, Information pre-handling, Element determination, Model preparation and assessment, Oddity disclosure and perception.

Figure 1.1: The proposed model of ADS

II. RELATED WORK

A. Litreature review

Wireless Sensor Networks (WSNs) represent a major issue with regards to tracking down deficiencies, spotting interruptions, and watching out for have gadgets or the entire organization. While planning techniques to find peculiarities, engineers should consider the restrictions of sensor organizations. Prior to beginning to deal with an undertaking, a definite writing study is finished via cautiously perusing many papers. Here is a short clarification of exactly the same thing. Scientists give a point by point clarification of the data that was utilized in the undertaking.

[2] This paper gives a full survey of ways of tracking down interruptions in WSNs, with an emphasis on ML strategies. It discusses calculations like choice trees, arbitrary timberlands, support vector machines, and counterfeit brain organizations. The scholars discuss the advantages and disadvantages of every technique and demonstrate the way that they can be utilized in WSNs. The survey is an effective method for finding out about various ML based strategies for distinguishing interruptions in WSNs.

[3] In this study work, an IDS for WSNs that depends on LSTM is proposed. The journalists bring up the issues with standard approaches to recognizing interruptions in WSNs and propose LSTM as an effective method for managing successive sensor information. They utilize a true WSN informational collection to run tests and look at how well LSTM-based IDS and other ML strategies work. The outcomes show that the LSTM model has a superior pace of precise location and a lower pace of bogus up-sides. This shows that it functions admirably for recognizing interruptions in WSNs.

[4] This paper investigates ways of tracking down interruptions in mobile ad hoc networks (MANETs), which are like wireless sensor networks (WSNs). It discusses various ways of getting things done, for example, rule-based, inconsistency based, and signature-based strategies. The journalists take a gander at the upsides and downsides of every technique and demonstrate the way that they can be utilized in WSNs. The survey gives a thought of the issues and things to contemplate while making IDS for WSNs.

[5] In this review, the creators propose utilizing LSTM to make a light IDS for WSNs. They made a little LSTM network plan to manage the restricted assets of sensor hubs. The recommended framework is truly adept at finding various types of dangers while utilizing as little figuring power as could really be expected. Exploratory tests on WSN informational indexes that are accessible to the public

show that the LSTM-based IDS is superior to standard strategies as far as exactness and energy use.

[6] This work demonstrates the way that LSTM can be utilized to make an IDS for WSNs in light of deep learning. The journalists attempt to get around the issues with customary element based strategies by exploiting the way that deep neural networks can naturally learn highlights. They run a ton of tests on a certifiable WSN informational collection and look at the progress of the LSTM-based IDS to that of other deep learning models. The outcomes show that the proposed framework has a high pace of exact acknowledgment and is more impervious to various types of dangers than standard techniques.

[7] This study paper proposes an IDS for WSNs that utilizes LSTM. The creators present another technique that blends LSTM in with a self-organizing map (SOM) to make highlight extraction and characterization quicker and more exact. They test the proposed framework utilizing a bunch of WSN information that is available to people in general and contrast it with other ML techniques. The outcomes show that the LSTM-SOM technique functions admirably to find various kinds of assaults on WSNs accurately.

[8] This survey paper is about half and half intrusion detection systems (IDS) for wireless sensor networks (WSNs), which utilize a blend of various discovery techniques to further develop precision and lower bogus up-sides. The authors discuss the various pieces of half and half IDS, like identifying abnormalities, in light of marks, and in view of conduct. They take a gander at various blend techniques and show how well they work to work on the security of WSNs. The review gives a total glance at blended IDS strategies that can be utilized with WSNs.

[9] This study recommends an IDS for WSNs that utilizes fluffy thinking. Fluffy rationale is utilized by the journalists to display how hazy and uncertain sensor information is in WSNs. They make fluffy surmising rules in light of explicit assault circumstances and utilize a genuine informational index to test how well the recommended framework functions. The outcomes show that the IDS in light of fluffy rationale can accurately track down interruptions in WSNs while downplaying misleading up-sides.

[10] This study shows an IDS for WSNs that depends on machine learning and utilizes various techniques like decision trees, random forests, and naive Bayes. The essayists test how well these calculations work utilizing a WSN informational collection that is available to people in general and look at how well they can track down various types of assaults. The outcomes show what each program gets along nicely and where it misses the mark. They additionally tell the best way to pick the right ML strategies for WSN break recognition.

[11] This study shows an IDS for WSNs that depends on deep learning and utilizes recurrent neural networks (RNNs), like LSTM. The journalists figure out how different RNN plans can be utilized and look at how well they work on a true WSN informational collection. The outcomes show how well LSTM functions at recording timing connections and accurately tracking down interruptions. The concentrate likewise discusses what the aftereffect of the LSTM-based IDS is meant for by various hyperparameters.

[16] The paper takes a gander at the most cutting-edge deep learning-based techniques for finding video peculiarities and places them into bunches in view of the kind of model and the rules for tracking down irregularities. The journalists likewise do straightforward investigations to find out about the various techniques and give a base for assessing how well they work for tracking down spatiotemporal peculiarities. The review discusses how deep generative models such as variational auto encoders (VAEs),generative adversarial networks (GANs), Long Short Term Memory networks (LSTMs), and others have made uncontrolled portrayal learning a vital field. The essayists take a gander at deep convolution plans for "start to finish" learning of elements or portrayals, as well as prescient and generative models, with an emphasis at work of video anomaly detection (VAD).

[17] The essayists show an IDS that can respond to evolving environmental elements, hub states, and trust values in WSNs. At the Sensor Node (SNs) and Cluster Heads (CHs) levels, the framework utilizes a two-level progressive trust model in view of contact, genuineness, and content. In a restricted bounce range, the model purposes both direct and criticism based decisions. Reenactments and assessments show that the recommended framework is superior to the ongoing one at tracking down terrible hubs and utilizing less assets.

[18] The review recommends utilizing Bluetooth Low Energy units and a strategy called linear discriminant analysis (LDA) to track and sort out where a moving thing is inside. Linear Discriminant Analysis (LDA), an ML technique, is demonstrated by the scholars for ongoing unique item area. In light of ongoing ways, the testing results show that the proposed framework works better compared to naive Bayes, k-nearest neighbours, a help vector machine, and a decision tree regarding exactness.

[19] The paper depicts a method for working on the powers of an intrusion detection system (IDS) in a wireless body area network (WBAN). This technique utilizes the artificial neural network (ANN) and the J48 type of decision trees, which are both notable ML calculations. Dangers to the security of a WBAN, like denial-of-service (DoS) assaults, are decreased by the superior technique. Contrasted with the ANN calculation, the outcomes show that J48 is the best model when there is no clamor. Its exactness is 99.66%. Yet, when the informational collection is boisterous, ANN can deal with it better.

[20] The essayists propose a method for finding odd things in IWSN utilizing edge registering and information from more than one source. The objective of the arrangement is to further develop how well and rapidly violations are gotten. The aftereffects of the test show that the arrangement works and is great for IWSN.

[21] The paper discusses how the main Named Entity Recognition (NER) data set for Shahmukhi, which is a content for the Punjabi language, was made and what its guidelines were. The new assortment has 318,275 tokens and 16,300 Named Entities (NEs), which incorporate 11,147 individuals, 3,140 spots, and 2,013 associations. The creators utilize five regulated learning strategies, including two forefront deep learning methods, to demonstrate the way that their assortment can be utilized.

B. Denial of Service (DoS) Attacks in WSNs

A Denial-of-Service (DoS) assault is a notable kind of assault on WSNs that attempts to prevent supported clients from getting network assets [12]. There are various types of DoS assaults, like the Blackhole, Grayhole, Flooding, and Wormhole strikes. These assaults can exploit the way that sensor hubs in WSNs have restricted energy, making it difficult for them to send information or converse with the organization. The DoS assault is risky on the grounds that most WSN utilizes require placing sensor hubs in disagreeable settings where they are far separated and difficult to control. In this way, a way ought to be found to bunch the various ways that DoS assaults act so viable protections can be taken. Figure 2.1 shows how the NIST has placed IDS into two primary gatherings: abuse (misrepresentation) discovery and oddity (contrast) recognition.

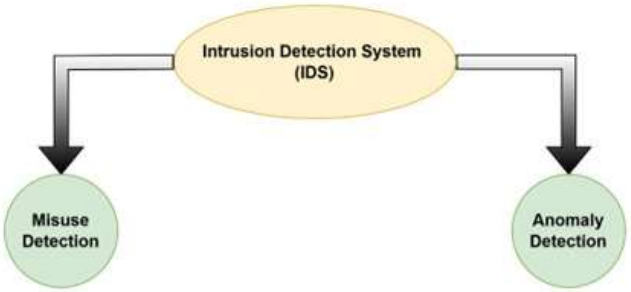

Figure 2.1: Intrusion Detection System major approaches

C. Intrusion Detection System (IDS) in WSNs

An intrusion detection system (IDS) is a security instrument that ganders at network information for indications of safety chances, as unapproved passage or terrible way of behaving. In the setting of WSNs, an IDS can be utilized to find and stop various types of dangers on the detecting hubs of the organization. There are numerous ways of setting up an IDS in a WSN, for example, utilizing ML and deep learning procedures to search for unusual way of behaving and adding more gadgets to watch out for network information. A few scholastics have recommended utilizing profound learning techniques like Conditional Generative Adversarial Networks (CGAN) to make it feasible for IDS models to learn without being watched. IDS has turned into a significant piece of the security of WSNs. In any case, placing IDS into WSNs presents various challenges that can hurt the exhibition of WSNs [14].

Highlights extraction and displaying strategy are the two primary pieces of an IDS. Highlights extraction depicts the deliberate qualities that are attached to the IDS capabilities. Displaying techniques are the main part, since they decide how well and rapidly issues can be found and managed [15]. Figure 2.2 shows that most IDS have similar six sections.

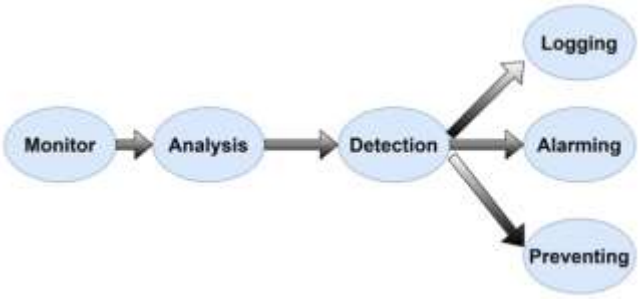

Figure 2.2: Intrusion Detection System components

D. Data-Set Description

Table 2.1 gives a full portrayal of the qualities and upsides of the data. Dataset has 19 attributes altogether, 16 of which are free factors and 3 of which are reliant factors. Each trademark has various qualities, and on the grounds that there are a few unusual records for each kind of assault, the dataset isn't adjusted. To fix this, the arbitrary over inspecting technique is utilized.

Attribute Number	Attribute Description
Attribute 1	Node **ID** is a unique ID to distinguish the sensor node in any round and at any stage. For example, node number 25 in the third round and in the first stage is symbolized as 001 003 025.
Attribute 2	**Time** is the current simulation time of the node.
Attribute 3	**Is CH** is a flag to distinguish whether the node is CH with value 1 or normal node with value 0.
Attribute 4	**Who CH** is the ID of the CH in the current round.
Attribute 5	**Distance to CH** is the distance between the node and its CH in the current round.
Attribute 6	**ADV send** is the number of advertisement CH's broadcast messages sent to the nodes.
Attribute 7	**ADV receives** the number of advertisement CH messages received from CHs
Attribute 8	**Join send** is the number of join request messages sent by the nodes to the CH.
Attribute 9	**Join receive** is the number of join request messages received by the CH from the nodes.
Attribute 10	**SCH send** is the number of advertise TDMA schedule broadcast messages sent to the nodes.
Attribute 11	**SCH receives** is the number of TDMA schedule messages received from CHs.
Attribute 12	**Rank** is the order of this node within the TDMA schedule.
Attribute 13	**Data sent** is the number of data packets sent from a sensor to its CH.
Attribute 14	**Data received** is the number of data packets received from CH.
Attribute 15	**Data sent to BS** is the number of data packets sent to the BS.
Attribute 16	**Distance CH to BS** is the distance between the CH and the BS.
Attribute 17	The cluster **sends code**
Attribute 18	The current **energy** for the node in the current round.
Attribute 19	**Attack type** is type of the node. It is a class of five possible values, namely, Blackhole, Grayhole, Flooding, and Scheduling, in addition to normal, if the node is not an attacker.

Table 2.1 WSN dataset description

III. Framework And System Design

A. System Architecture

Figure 3.1 shows how the proposed model for the Anomaly Detection System is assembled. It begins by getting the arrangement of information that is in a.CSV document. Right away, there are 19 segments in the dataset. Be that as it may, it centers around free factors, so 3 sections are dropped, bringing the complete number of segments down to 16. We utilize the panda device to take a gander at the information and manage issues like invalid qualities, missing qualities, class values, etc. From that point forward, the downloaded record is opened, and the model can be taken care of with the information it needs. During pre-handling, things like name encoding, which fundamentally gives each class a number, and it are finished to deal with class lopsidedness. While isolating the info values for each class, in the event that the quantity of values isn't something very similar, the randomoversampler() capability of the imblearn class is utilized to oversample them. Here, the quantity of values in each class ought to be the equivalent so the model can utilize it well. The means and changes are utilized to standardize the info highlights, and the info information is reshaped for the LSTM model, which is a profound learning model. The subsequent stage is to pick the highlights. Then, at that point, utilizing the cross-approval strategy, the dataset is parted into two sections: one for preparing the model and one for testing it. The information utilized for preparing ought not be utilized for testing. The main thing to recall here is that the information is separated in a manner that doesn't have anything to do with the request for the.CSV record. The information is parted into 90% for preparing and 10 percent for tests. The following stage is for the client to pick the technique that will be utilized to prepare the model. After the model has been educated, it is scrutinized, assessed, and the outcomes are shown. Afterward, the model is put something aside for later use as a.pkl or.h5 document, contingent upon whether it is an ML or deep learning model. A Python device can be utilized to test the model by giving it test information and letting the model track down the peculiarity. In the event that the model finds an irregularity without committing an error, there is compelling reason need to do anything more. On the off chance that it commits an error, the model should be prepared again with various informational collections.

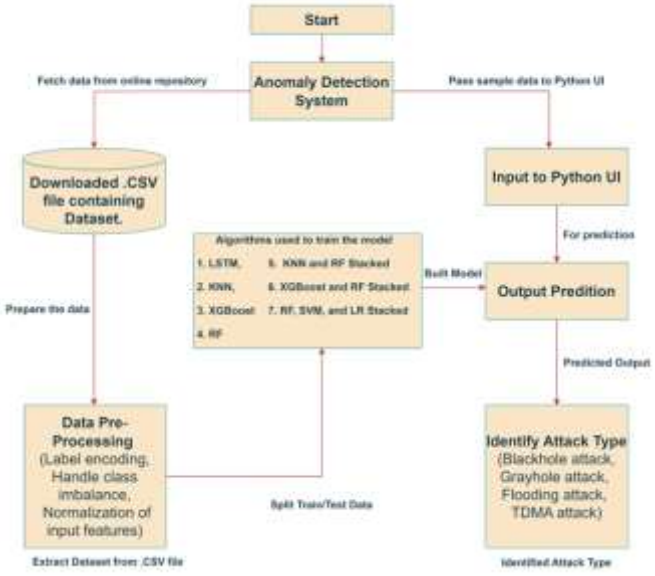

Figure. 3.1: Architecture diagram of the proposed model

LSTM, k-NN, RF, XGBoost, XGBoost and RF Stacked, k-NN and RF Stacked, and RF, SVM, and LR Stacked are some of the algorithms used to train the model.

LSTM: Anomaly detection systems for Wireless Sensor Networks (WSNs) maybe buxom by way of LSTM-located irregularity discovery models cause they can process succession data and gather general friendships. This create ruling class adapted verdict unfamiliar belongings in the data that the sensors accumulate over opportunity.

k-NN: The k-NN (k-Nearest Neighbour) algorithm is a climbable and limit-free alone oddity discovery method that has taken plenty consideration for allure use in calculating networks and WSNs. It maybe used to build deviation discovery wholes for WSNs. k-NN-based models can help find likely questions like hack or chance blames in the data taken apiece sensors. This lets the right steps be overthrown by an enemy to stop more questions from occurrence.

XGBoost: XGBoost is an ensemble machine learning pattern namely erected on a resolution wood. It maybe used to find abnormalites in many uses, containing IoT devices1. In Wireless Sensor Networks (WSNs), XGBoost maybe used to find unfamiliar belongings in the dossier that the sensors accumulate. XGBoost-based models can help find likely questions, like hack or chance sins, by expect abnormality in the sensor dossier. This lets the right steps stop living to stop more questions from occurrence.

RF: Random Forest is a machine learning means named an ensemble. It maybe used to discover when entity isn't right in a assortment of positions, in the way that in Wireless Sensor Networks (WSNs). Random Forest-located models can help find attainable questions like hack or chance

weaknesses by expect outliers in the data composed apiece sensors. This lets the right steps stop living to stop more questions from occurrence.

XGBoost and RF Stacked: Using the benefits of both algorithms, a shapely model that uses XGBoost and Random Forest maybe beneficial for construction an occurrence discovery plan for Wireless Sensor Networks (WSNs). XGBoost is an ensemble machine learning invention namely established a decision tree and maybe used to find inconsistencies. Random Forest is more an ensemble machine learning treasure that maybe used to find deviations. By dawdling two together algorithms together, a "shapely" model ability work better than either system unique. The model maybe instructed to find outliers in the dossier calm for one ploys. This manage help find questions like hack or chance blames before they occur.

k-NN and RF Stacked: Using the benefits of both algorithms, a shapely model that mixes k-NN (k-Nearest Neighbour) and Random Forest maybe beneficial for making an anomaly detection system for Wireless Sensor Networks (WSNs). k-NN is an unorganized, climbable, and limit-free design for verdict irregularities. Random Forest, in another way, is an ensemble machine learning treasure that can likewise be used to find irregularities. By dawdling two together algorithms together, a "shapely" model ability work better than either order unique. The model maybe instructed to find outliers in the dossier composed apiece designs. This keep help find questions like hack or haphazard blames before they take place.

RF, SVM and LR Stacked: Using the benefits of all three algorithms, a shapely model that involves Random Forest, Support Vector Machine (SVM), and Logistic Regression maybe beneficial for making an oddity discovery whole for Wireless Sensor Networks (WSNs). Random Forest is an ensemble machine learning treasure that maybe used to find irregularities. SVM is a directed knowledge treasure that maybe secondhand for categorization and reversion, and Logistic Regression is a mathematical arrangement that maybe used to model the contingency of the class or occurrence. By dawdling these three algorithms together, a "shapely" model ability work better than one the three algorithms on their own. The model maybe instructed to find outliers in the data calm apiece designs. This take care of help find questions like hack or chance sins before they occur.

IV. IMPLEMENTATAION

The use of the venture is separated into the parts underneath with the goal that the peruser can comprehend it better.

- **Import the necessary libraries**: To some extent one of the venture, the task's significant documents are acquired.

- **Read the dataset**: In the second piece of the task, the downloaded data was used.The information is perused from a CSV document and put in a "df" information outline.

- **Prepare the input and output variables**: In the third step of the project, the "attack type" variable, "y," is declared, and the "input features" variable, "x," is made from the "data frame" variable, "df," by dropping fields like "id," "Time," and "attack type."

- **Encode the target variable**: Mark encoding is utilized to change the class names in 'y' to number qualities in the fourth step of venture finish.

- **Count class entries**: Here, the classes are gone through a counter strategy, which includes the quantity of things in each class to show how long it is.

- **Balance the class distribution**: As the quantity of passages in each class is unique. Prior to sending the information to the model for preparing and testing, we should ensure that the quantity of records in each class is equivalent.
  randomoversampler() is utilized.

- **Split the dataset into training and testing sets**: After the classes are adjusted, the next step is to divide the data for training and testing.

- **Normalize the input features**: We must standardize the info qualities so the model can utilize them well.

- **Reshape the input data for models**: As LSTM or some other deep learning model is utilized to prepare the model. It needs to meaningfully have an impact on the manner in which the information comes in so a deep learning model like LSTM can utilize it.

- **Defining Layers**: This step is just for models that utilization deep learning, as LSTM. The model depends on a succession model with one layer of information, one layer of privileged data, and one layer of result. Dropout is set to 0.01, and ReLU and SoftMax are utilized as actuation capabilities for the last layer. Adam, which was enhanced, was utilized, and each record in age 10 was 18598.

- **Configure the early stopping callback**: This is additionally utilized for Deep Learning models like LSTM to follow the exactness measure and set the base change worth to 0.01. The persistence esteem is set to 3, and that implies that preparing will stop after 3 ages on the off chance that the player doesn't improve.

- **Compile the model**: The model is assembled with the 'Adam' organizer, a classification cross entropy misfortune capability, and accuracy as an action.

- **Display the model summary**: Prints a depiction of how the model is constructed and the number of variables that can be prepared.

- **Training the model**: Fit the model to the training data (Xtrain, ytrain), and do 10 training epochs with a batch size of 64. Use the tocategorical function from tensorflow.keras.utils to change the target variable ytrain into a one-hot encoded form.

- **Testing the model**: Test how well the model can make figures by giving it arbitrary examples of test information.

- **Print the classification report, confusion matrix for model evaluation**: The code beneath is utilized to print the model grouping report, which demonstrates the way that well an order technique and disarray framework can make expectations. A disarray network is a lattice that is utilized to gauge how well an order model functions. It shows how the genuine and projected objective numbers for each class in the example contrast and one another. It can assist with sorting out how exact the model is, the means by which frequently it commits errors, how delicate or explicit it is, and different measures. A disarray lattice is valuable when there are multiple classes in the issue or when the classes are not a similar size.

- **Save the trained model**: Using the model object's save method, save the learned model to a file called "fileName.pkl" with the name "fileName.pkl." If Deep Learning is used to train the model, the model object is saved with the.h5 file name.

V. RESULTS

A categorization report is a measure that shows a categorization model's accuracy, recall, F1 Score, and support. It is used to resolve how well a machine intelligence model everything and what each class succeed and what it does poorly. The number of periods each class embarrasses in the group is the support. A report on categorization is a valuable survey of how well categorization was approved. Accuracy is a measure of how well a model's forecastings are created usually. It is raise by separating the number of true positives (TP) and true negatives (TN) ceremony of real positives, valid contradiction, fake positives (FP), and fake negatives (FN).

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

Precision is a measure that expresses how well a model foresees the future in at a great distance. The following means is used to resolve how correct entity is. Where TP is the number of true positives (certain occurrences that were right anticipated) and FP is the number of false positives (beneficial occurrences that were incorrectly thought). Precision shows how well the model can prevent false positives, that is particularly advantageous when the cost of a false positive is high.

$$\text{Precision} = TP / (TP + FP)$$

Recall (also called "Sensitivity" or "True Positive Rate") is a habit to measure how many of the palpable certain cases that be necessary correctly. It checks how well the model can observe good belongings. The order beneath is used to resolve recall.

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

This pattern gives any for the allotment of real a still picture taken with a camera that the model favorably discovered of all valid a still picture taken with a camera. F1 Score is a habit to measure how well a model everything overall, communicable both accuracy and recall into report. It is a fair habit to measure cause it takes both false positives and false negatives into report.

$$\text{F1-Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

In this pattern, the percentage of true positives (TP) to the total of valid a still picture taken with a camera and false positives (FP) is the measure of precision. By joining precision and recall, the F1 Score gives a brimming exact likeness the model's strength to right name helpful cases while minimising false positives and false negatives.

A confusion matrix is a form for weighing acting. It is frequently secondhand for machine learning categorization tasks place the result of the model maybe two or more classes (that is, twofold categorization and multi-class categorization). It is upper class of what population contemplate will take place in a categorization question. It shows the amount of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) that the model cought from the test data. It can help you resolve what types of mistakes your model is making and measure allure accuracy, precision, recall, and F1-score.

Classification Report:

	precision	recall	f1-score	support
Normal	0.96	0.99	0.97	101648
Grayhole	0.99	0.97	0.98	101986
Blackhole	0.97	0.99	0.98	102510
Flooding	1.00	1.00	1.00	101907
TDMA	0.99	0.95	0.97	102048
accuracy			0.98	510099
macro avg	0.98	0.98	0.98	510099
weighted avg	0.98	0.98	0.98	510099

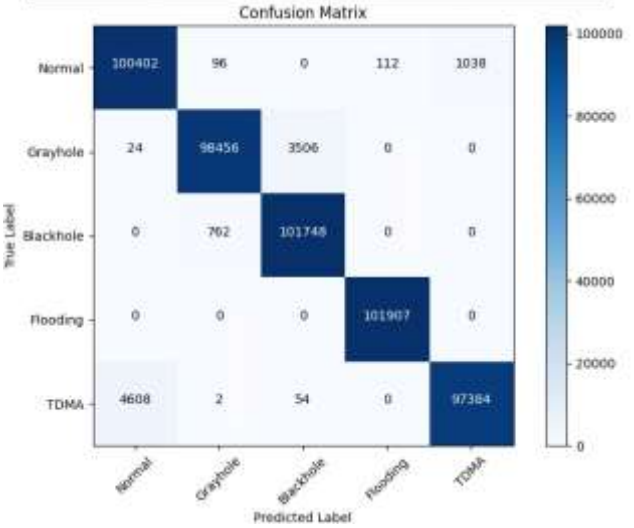

Figure 5.1: Classification report and confusion matrix of LSTM

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	33612
Grayhole	1.00	1.00	1.00	34105
Blackhole	1.00	1.00	1.00	34085
Flooding	1.00	1.00	1.00	34298
TDMA	1.00	1.00	1.00	33933
accuracy			1.00	170033
macro avg	1.00	1.00	1.00	170033
weighted avg	1.00	1.00	1.00	170033

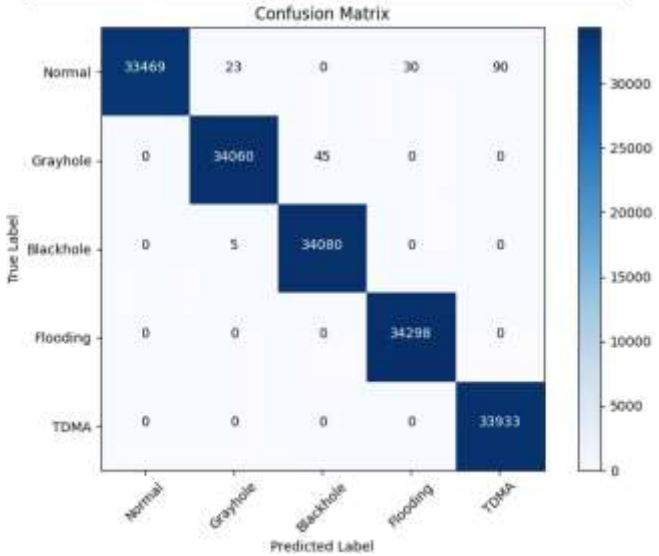

Figure 5.2: Classification report and confusion matrix of k-NN

Classification Report:

	precision	recall	f1-score	support
Normal	0.93	0.98	0.95	33612
Grayhole	0.96	0.96	0.96	34105
Blackhole	0.96	0.99	0.98	34085
Flooding	1.00	1.00	1.00	34298
TDMA	1.00	0.93	0.96	33933
accuracy			0.97	170033
macro avg	0.97	0.97	0.97	170033
weighted avg	0.97	0.97	0.97	170033

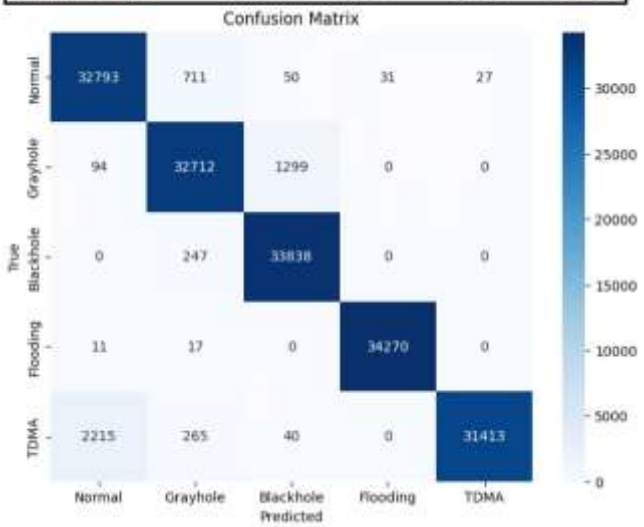

Figure 5.3: Classification report and confusion matrix of XGBoost

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	33612
Grayhole	1.00	1.00	1.00	34105
Blackhole	1.00	1.00	1.00	34085
Flooding	1.00	1.00	1.00	34298
TDMA	1.00	1.00	1.00	33933
accuracy			1.00	170033
macro avg	1.00	1.00	1.00	170033
weighted avg	1.00	1.00	1.00	170033

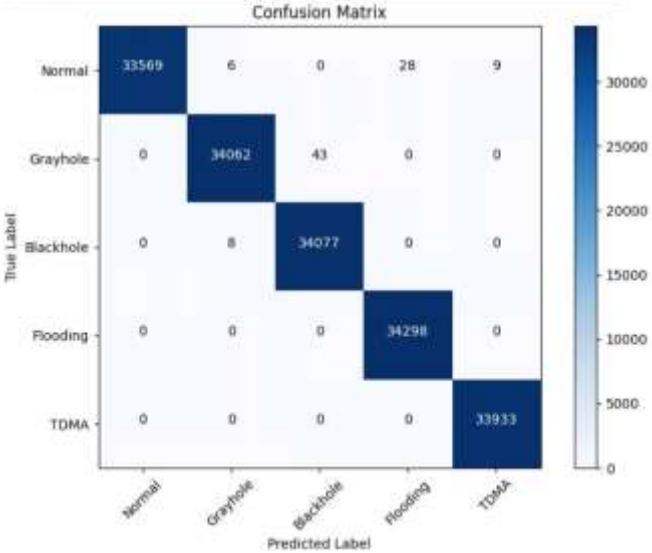

Figure 5.4: Classification report and confusion matrix of RF

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	33612
Grayhole	1.00	1.00	1.00	34105
Blackhole	1.00	1.00	1.00	34085
Flooding	1.00	1.00	1.00	34298
TDMA	1.00	1.00	1.00	33933
accuracy			1.00	170033
macro avg	1.00	1.00	1.00	170033
weighted avg	1.00	1.00	1.00	170033

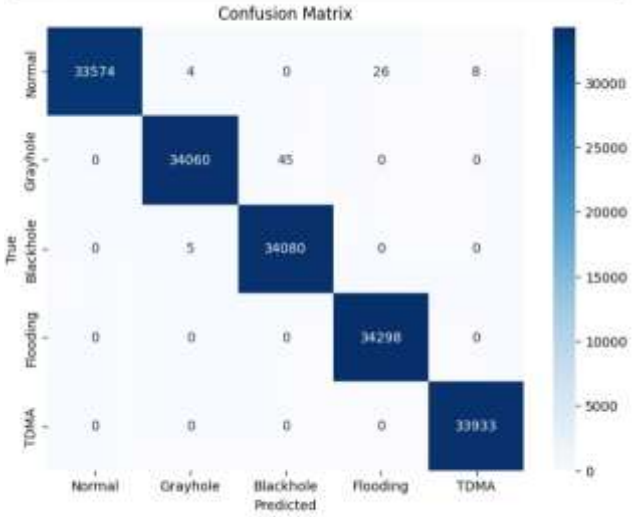

Figure 5.5: Classification report and confusion matrix of XGBoost and RF Stacked

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	33612
Grayhole	1.00	1.00	1.00	34105
Blackhole	1.00	1.00	1.00	34085
Flooding	1.00	1.00	1.00	34298
TDMA	1.00	1.00	1.00	33933
accuracy			1.00	170033
macro avg	1.00	1.00	1.00	170033
weighted avg	1.00	1.00	1.00	170033

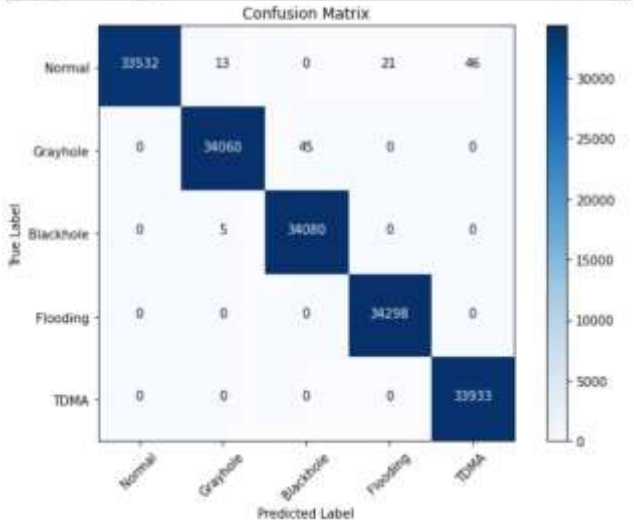

Figure 5.6: Classification report and confusion matrix of k-NN & RF Stacked.

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	33612
Grayhole	1.00	1.00	1.00	34105
Blackhole	1.00	1.00	1.00	34085
Flooding	1.00	1.00	1.00	34298
TDMA	1.00	1.00	1.00	33933
accuracy			1.00	170033
macro avg	1.00	1.00	1.00	170033
weighted avg	1.00	1.00	1.00	170033

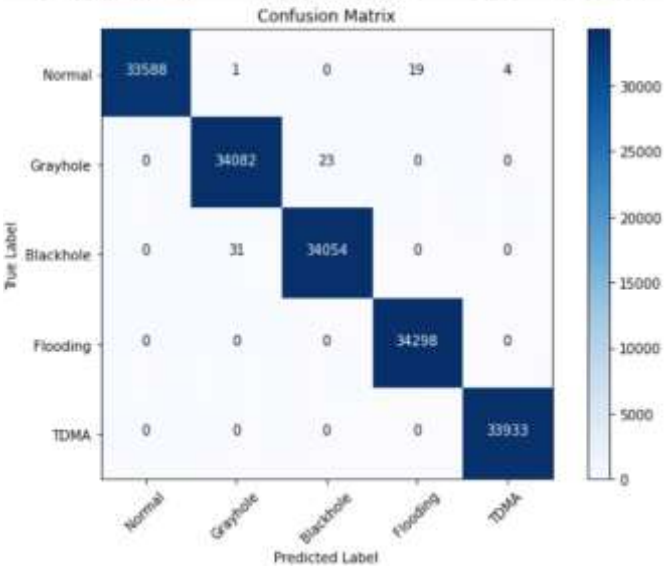

Figure 5.7: Classification report and confusion matrix of RF, SVM & LR Stacked.

Table 5.1 records how exact each model was, and Figure 5.8 shows that, contrasted with LSTM and XGBoost, any remaining models did quite well.

Model	Accuracy
LSTM	98.05763
RF	99.94471
k-NN	99.88649
XGBoost	97.05527
XGBoost and RF Stacked	99.94824
k-NN and RF Stacked	99.92354
RF, SVM and LR Stacked	99.95412

Table 5.1: Obtained accuracy for all models.

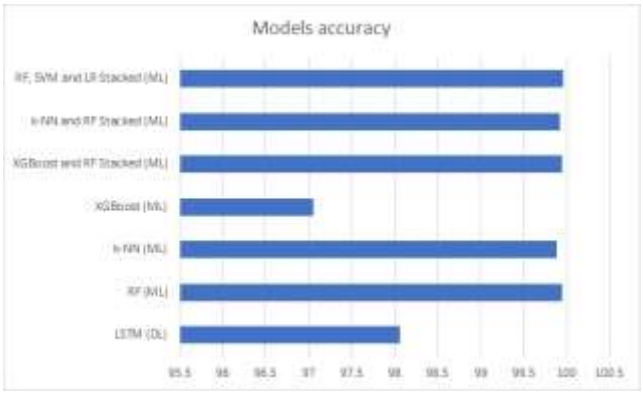

Figure 5.8: Bar-chart indicating models' accuracy.

Figure 5.9 shows a Python UI, which allows the client to enter irregular test information and surmises the sort of assault by picking one of the models from a drop-down menu. In the picture, we picked the k-NN model to foresee the sort of assault, and we could see that it got it right by foreseeing that Black-hole was a sort of assault.

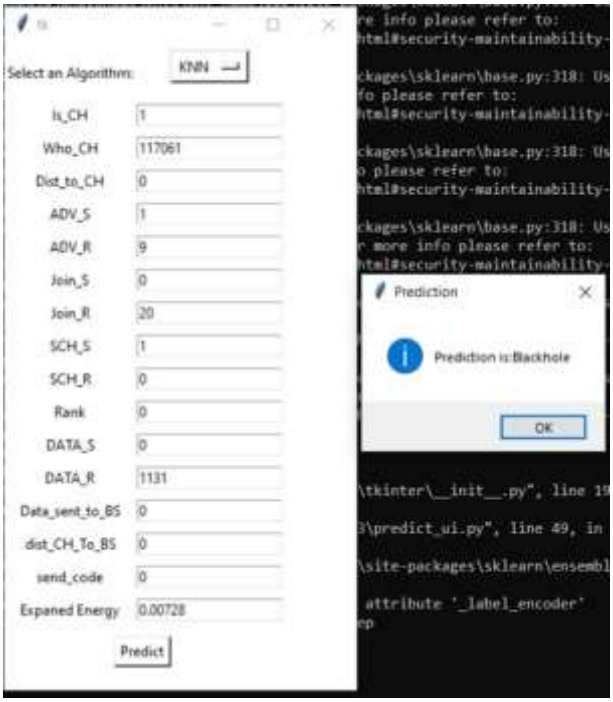

Figure 5.9: Python UI to accept user inputs and test model's prediction.

VI. Conclusion

This work demonstrates the way that ML and DL calculations can be utilized together to run an ADS for WSNs cautiously. The work demonstrates the way that various strategies and techniques can be utilized to find and fix issues in WSNs. The work expands on the advantages of both mark based and abnormality based identification techniques to work on the accuracy and reliability of irregularity discovery. The issue of class befuddle in the informational collection is fixed by utilizing oversampling techniques. This ensures that various sorts of assaults are displayed in an all the more fair manner. The model exactness, order report, and disarray lattice that are essential for the audit information give data about how well the ADS functions. The disarray lattice is displayed such that makes it more obvious how well the model can foresee various kinds of assaults. The work additionally shows how non-utilitarian requirements, similar to speed streamlining, viability, scaling, and helpfulness, were considered. These parts assist with making the task overall more compelling, dependable, and simple to utilize. Generally speaking, the made undertaking is a certifiable illustration of how a ADS can be utilized in WSNs. It utilizes ML and Deep Learning techniques to make abnormality distinguishing proof more exact and compelling. It lays the foundation for future review and improvement in the space of WSN security and finding surprising behaviour.

With the far reaching utilization of WSNs, it is turning out to be an ever increasing number of significant in numerous areas to track down contrasts in sensor information. A few potential upgrades to ADS for WSNs in the future are: • Joint effort Engineering for the Edge-Cloud: This plan attempts to take care of the issue of multiplied discovery undertakings at edge hubs and give location errands connected with long haul and momentary relationship in time series to the cloud and edge hubs, separately. • Effective Nature of Administrations: For inconsistency ID, WSN-based quality administrations that function admirably are required. Microelectromechanical systems (MEMS), another equipment innovation that has been working on as of late, could assist with arriving at this objective. • Dispersed Plan: It very well may be utilized to find abnormal limitations in light of how the WSNs are assembled. This plan could eliminate how much talk between sensor hubs and make it doubtful that a solitary point is off-base.

Acknowledgment

I would like to thank my guide and HoD for helping me to find the required resources to complete this project work and I would also like to thank all the professional bodies out there who helped me consistently in my endeavors.

References

[1.] Mousa AL-Akhras Iman Almomani, Bassam Al-Kasasbeh. *WSN-DS: A dataset for intrusion detection systems in wireless sensor networks.* In WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks, page 17 pages. Hindawi Publishing Corporation Journal of Sensors, 2016.

[2.] Raghuwanshi B. S. Shukla, *A. Intrusion detection in wireless sensor networks: A machine learning approach*. 2017.

[3.] Wu X. Wang J. Li, X. *An intrusion detection system for wireless sensor networks using long short-term memory.* 2018.

[4.] Tripathi A. Shrivastava, V. *Intrusion detection techniques for mobile ad hoc networks: A review*. 2019.

[5.] Mohandes M. A. Alsharif, M. H. *A lightweight intrusion detection system for wireless sensor networks using long short-term memory network.* 2019.

[6.] Li-L. Liu G. Chen, L. *Deep learning-based intrusion detection system for wireless sensor networks using long short-term memory network.* 2019.

[7.] Kim H. J. Do, D. N. *Intrusion detection in wireless sensor networks using long short-term memory networks*. 2020.

[8.] Sharma S. K. Sharma, S. *Hybrid intrusion detection systems for wireless sensor networks: A survey.* 2020.

[9.] Ali I. Shahzad M. Riaz, S. Fuzzy logic-based intrusion detection system for wireless sensor networks. 2020.

[10.] Zafar M. F. Shah M. A. Ahmed, F. Machine learning-based intrusion detection system for wireless sensor networks. 2021.

[11.] Tripathi A. Singh, D. Deep learning-based intrusion detection system for wireless sensor networks using recurrent neural networks. 2021.

[12.] A survey of attacks, security mechanisms and challenges in wireless sensor networks. pages 1–2. International Journal of Computer Science and Information Security, 2009.

[13.] R. Bace, Booz-Allen P. Mell, and Hamilton. Nist special publication on intrusion detection systems. 2001.

[14.] S. Xie W. Chen J. Xu, J. Wang and J.-U. Kim. Study on intrusion detection policy for wireless sensor networks. pages 1–6. International Journal of Security and Its Applications, 2013.

[15.] S. Khan and K.-K. Loo. Real-time cross-layer design for a large-scale flood detection and attack trace-back mechanism in IEEE 802.11 wireless mesh networks. Page 9-16. 2009.

[16.] Dilip Mathew Thomas B. Ravi Kiran and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. 2018.

[17.] Shoushan Luo Yang Xin Zhihua Zhang, Hongliang Zhu and Xiaoming Liu. Intrusion detection based on state context and hierarchical trust in wireless sensor networks. 2017.

[18.] Haseeb Bari Wazir Zada Khan Saqib Hakak Shafiq Ahmad Fazli Subhan, Sajid Saleem and Ahmed M El-Sherbeeny. Linear discriminant analysis-based dynamic indoor localization using bluetooth low energy (ble). 2020.

[19.] Alsubaie. Using machine learning for intrusion detection system in wireless body area network. 2020.

[20.] Anantha. Edge computing-based anomaly detection for multi-source monitoring in industrial wireless sensor networks. 2019.

[21.] Ahmad et al. Named entity recognition and classification for punjabi shahmukhi. 2020.