Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

LOCALIZING IOT DEVICES IN INDOOR ENVIRONMENT USING AI-DRIVEN CLOUD FRAMEWORK

Sourav Kumar Bhoi[1], Kalyan Kumar Jena[1*], Ashish Kumar Mohanty[1], Monalisa Kar[1]
[1]Department of Computer Science and Engineering, Parala Maharaja Engineering College, Berhampur-761003, India Email: sourav.cse@pmec.ac.in, kalyan.cse@pmec.ac.in*, ashishmohanty@outlook.com, monalisakar99@gmail.com

Abstract:
Localization of IoT devices refers to finding out the exact position of IoT devices (Mobile, Wi-Fi, etc.). Many different localization algorithms, techniques and methodology are developed for both indoor and outdoor environments. However, indoor localization is still an open problem because of limitation of Global Positioning System (GPS) and Radio signals in indoor environment. Therefore, user localization is an important research concern in the current situation. In this work, the localization technique to find the exact position of IoT devices in an indoor setting will be focused on using the Supervised Machine learning methods. Here the classification models are used such as K-Nearest Neighbors (kNN), Support Vector Machine (SVM), and Random Forest (RF) will be used on a predefined data set to note the precision of the localization and to predict the floor and Room ID. The data set for this work is taken from the University of California, Irvine (UCI). This work will be evaluated using the performance parameters like accuracy, precision, recall and support and it will be implemented using Python programming languages using libraries like Pandas, NumPy, SkLearn. From the above models result, it has been observed that kNN model show better result with 98% accuracy. A hardware experiment using smartphones is also conducted at last to validate the work.
Keywords: Machine Learning, IoT Localization, Supervised Machine Learning, K-Nearest Neighbors, Accuracy, Precision

1. Introduction
Localization means finding the location of smart devices using many new localization techniques and methods [1-34]. In now era millions of IOT device are used. These IOT device intelligently connect to form smart devices. Generally, localization is 2 types (i) Outdoor Localization (ii) Indoor localization. In outdoor localization, there exists GPS system for finding the position of object or persons or things. But finding the location of any things or any person is only possible when it is outside any building. In contrast to outdoor positioning, finding location of anything like object or person inside the building or in a closure area whose barrier cannot be possible. In order to find that type of location of anything or person in closed indoor environment. Indoor environment includes Libraries, airport, colleges, schools etc. GPS systems cannot be used for finding these smart devices in indoor locations because the signals of the satellites are blocked by hurdles like roofs and walls of an indoor environment. Now a days 80% people depend upon smart devices. For finding these smart devices mainly in smart cities are challenging. In order to achieve this supervised machine learning model is being used.
As we know conventional algorithms takes some data and logic in form of code as input and gives output after processing it based on some predefined rules. These rules are made by some human experience on frequently-occurrence scenario.it take massive amount of investment, if the number of scenarios is increased and so to define rule it will going to difficult and hence it will sacrifice its accuracy and efficiency so here traditional algorithm is not used. However, in a Machine Learning (ML) algorithm, input and its predicted output are taken as the input again which produces some logic as output which will later be used as an input to make a further new output and so on. It is accuracy and efficiency does not depend on external factor, so machine learning model is being used.

By using supervised machine learning, it is four classifier model named as (i) kNN, (ii) SVM, (iii) Random Forest, (iv) Logistic Regression and one ensemble model named as voting classifier being used to determine the localization of IOT device.
In this paper, raw RSSI values are taken from the dataset [21] processed and splitted into two different datasets namely training dataset and testing dataset. Then different machine learning algorithms namely kNN, SVM, and RF put in our ML models to predict the location of different IoT devices (Android phones). The results of the respective models are compared to find the best one among them. The comparison the results of Localization of IoT devices following method are done based on Precision value, Recall value, F1-score value, and Accuracy values of ML models. Then to get a better result from all the above models' hybridization is done using voting classifier model.
This paper tires to contribute in following ways:

- Different models namely kNN, SVM, Random Forest, Logistic Regression are used for the localization of different IoT devices in indoor environment. The resultant outputs are then set side by side to each other to find out the best algorithm for the solution. The results are compared to each other based on Precision value, Recall value, F1-score value, and Accuracy values.
- Voting classifier is used as ensemble model to get the better result than all the other ML models based on Precision value, recall value, F1-score value, and Accuracy values and it is found that the voting classifier gets a better result.
- A deeper understanding of training model is known after doing an experimentation, where we make our own dataset of three different RSSI values and the class of the location of the IoT device. After which we put it in our ML models for training and get satisfactory results.
- From results it is found that kNN performs better in detecting the indoor IoT devices with an accuracy of 98%.
- A hardware experiment is also conducted in indoor environment using smartphones to validate the work successfully.

The remaining portion in our work are described as so, section 2 depicts about the related work where similar types of work is done by the author whom we mentioned in our reference. Section 3 describes the methodology where we show the uses of different supervised machine learning model with algorithms. Section 4 contains result and discussion part where performance of all models is assessed. Section 5 discuss about the hardware experiment using smartphones.

2. Related Work

The solutions on the problem of indoor localization are still lacking. However, many research works are conducted in this filed and some are discussed as follows. Shang Ma et al. [1] introduces a visible light-based positioning system for different IoT devices. They were able to do so by producing unique special encoding when the reflecting mirrors inside paperors were overturned.
Röhrig et al. [2] describe indoors tracking using a wireless network as this offers a cheap service for localization of IoT devices by using time-based range measurement. The major issues they faced were errors by multipath interference and NLoS signal propagation.
Bargh et al. [3] proposed an indoor localization technique using Bluetooth technology. It tracked the location of a motionless mobile. However, it had a restriction of being limited to a single room. It was infrastructure and network-based, so the mobile device didn't have to be customized for it to be localized.
Among the many different indoor positioning techniques, Wi-Fi is favored by many due to it not requiring the access points to be in line of sight of each other and therefore being suitable for complex indoor environment. He et al. [4] survey the advancement of two main techniques of indoor localization using Wi-Fi.
Moghtadaiee et al. [5] introduces a new localization method using frequency modulation (FM) broadcast. They considered the deterministic and probabilistic approaches and proposed a new method by combining the two approaches to improve the accuracy.

Shala et al. [6] examined the level of accuracy in localization by using built in sensors of an Android device. They focused on doing the localization of the android device in a room where the signal of the GPS was partially blocked or unavailable.
Otsason et al. [7] presented GSM based localization of indoor localization which reached the accuracy of 4m – 5m in a multistoried building. They conducted experiments on three different multi-floored buildings and reached to the conclusion that their system achieved accuracy comparable to a Wi-Fi based technique.
Liu et al. [8] investigated the suitability of Wi-Fi localization for accurate indoor localization of smartphones. They found from their results that considerable errors existed in this technique. They proposed a peer assisted localization technique to improve the results up to 80 percentiles.
Jeong et al. [9] introduced an indoor Localization Algorithm that was Smartphone-Assisted which they have named SALA. This algorithm showed the graphical display of IoT devices on a smartphone. A smartphone was used as a beacon which not only traced its own indoor position using its different sensors but also tracked other IoT devices by broadcasting close range pings and collected their information.
Bregar et al. [10] proposed two techniques are used to reduce the error in localization in an NLoS condition. The methods were based on ranging error regression model and NLoS classification model, both of them are CNN methods.
Bianchi et al. [11] presented a room-level localization using a novel RSSI-based fingerprinting technique. In the proposed technique the user location was estimated for their interaction with the home Wi-Fi system that was readily available.
Ouameur et al. [12] proposed a new foundation for indoor localization techniques in NLoS conditions which used low power wireless network. The framework used ML models and deep learning models to predict the location of different IoT devices.
Zhang et al. [13] introduces a new synchronous localization model using RFID for IoT devices which could locate objects at an accuracy of up to 30 centimeters. Their experimental study and real paper showed that the system proposed was superior.
Sadowski et al. [14] compared four different indoor localization methodologies which are Wi-Fi, Bluetooth, Zigbee, and long-range Wi-Fi network. These were then compared in relations to their respective accuracy and power consumption scores to decide which one of them we better.
Cai et al. [15] proposed that acoustic-enabled indoor localization solutions which garnered a lot of attention amongst other localization techniques are very costly due to the infrastructure and its maintenance. Also, this approach had much latency. To tackle these concerns, they presented a non simultaneous sound based localization system with participatory sensing.
Bhatti et al. [16] proposed that an outlier detection technique for localization which they named as IF-ensemble. They were successful in doing so by analyzing the RSSI value by using both supervised and unsupervised machine learning. They used research isolation forest as an unsupervised learning method. They used SVM, kNN and RF classifiers with stacking as supervised learning method.
Le et al. [17] examined the problem of indoor positioning of different IoT devices using the RSSI vectors without any prior awareness of the position of the IoT device. They use supervised machine learning method and modified KNN algorithm and proposed method to detect the unreliability of the predicted position.
Singh et al. [18] evaluated the best network requirements which resulted in minimal Average Localisation Error (ALE). They used a SVM model. They have proposed three different methods that were based on attribute similarity classification for rapid and correct paperion of ALE.
Huang et al. [19] introduces Kalman-filter drift removal and Heron-bilateration location estimation to remove errors in the RSSI value drift, positioning error, and reduce the installation cost of regular RFID indoor localization techniques. They accomplished these without any loss in the positioning accuracy.

Tran et al. [20] proposed VLC (visible light communication) positioning technique by combining KNN and RF for reflected environment. In that model which is based on fingerprint. he first treated KNN as a remedy for the increase in the amount of input features for RF. After that he finds the most effective one by ranking the input feature and the rest is removed to reduce computation effect. After that the model was trained. Finally, the process of estimation was used to find the proper position.
From the above study it is found that very less work done in establishment of a cloud-based framework for detection of indoor devices. In this work, a cloud-based framework is proposed to detect the IoT devices in indoor environment. The IoT devices are computationally limited, therefore this architecture is needed for performing the complex tasks like machine learning at the cloud. Also, hardware experiments are very rarely done for detection of IoT devices in previous works. In this work, it is performed to validate the proposed work.

3. Method and Materials

The methodology starts from the IoT network where the IoT devices are connected to the IoT network as per figure 1. The IoT device in this network can sense the physical environment. The devices are deployed in an indoor environment or the devices are dynamic in nature means it is moving from one indoor location to other. The IoT devices I={I1, I2,…., Im} sends continuous signals or beacons to the wi-fi devices W={W1,W2,….,Wn} connected in that indoor environment and m>>n where m is the number of IoT devices and n is the number of wi-fi devices in the indoor environment. The wi-fi device can send and receive data. It is connected to the IoT gateway that is connected to the Cloud C. The IoT gateway is responsible for sending data to IoT cloud and receiving the data from IoT cloud through base station. The IoT cloud is responsible for performing several applications and providing these services well to the users connected to it on demand. The IoT devices has limited computational capability whereas the IoT cloud has very large computational capability. It can perform complex operations for example machine learning for classification of IoT devices based on the RSSI (Received Signal Strength Indicator) values.

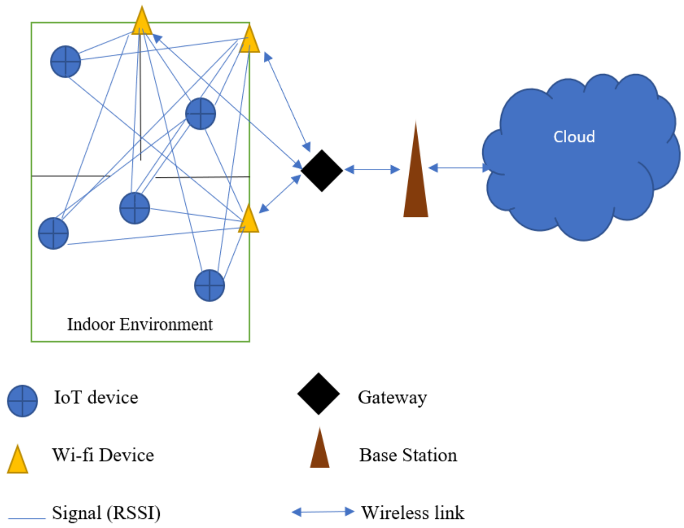

Figure 1: System architecture framework

The methodology starts as follows. The IoT devices are deployed static or dynamic in an Indoor environment such as office, industry, hall, etc. The devices continuously need to locate themselves using GPS however due to indoor environment the GPS fails to give services or localize the IoT devices whether they are in which floor, room, corridor, hall, etc. in an indoor environment. For this, the framework consists which continuously receives the IoT device signal S or beacon. The strength of this signal is extracted at the Wn and it is called as the RSSIn value. Then this RSSIn value is send to IoT gateway. Then the IoT gateway sends this value to cloud C. Then cloud C collects these RSSI values for an IoT device from n number of wi-fi devices. Then cloud C uses its best machine intelligence model to classify the series of RSSI value to a target label (room, area, zone, hall, corridor,

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

etc.). Here, we assume that the model is already trained using a standard dataset of the same indoor environment with z samples of data for training and testing. Here a sample contains n number RSSI values generated by the wi-fi devices for a device at any time t and a target label such as room number/ floor number/ corridor number/ hall number/ office number/ area number/ etc. Here, we have considered some supervised machine learning algorithms for training and testing to find the best model with high classification accuracy. The process of training and testing to find best model for the cloud is shown in figure 2 as follows.

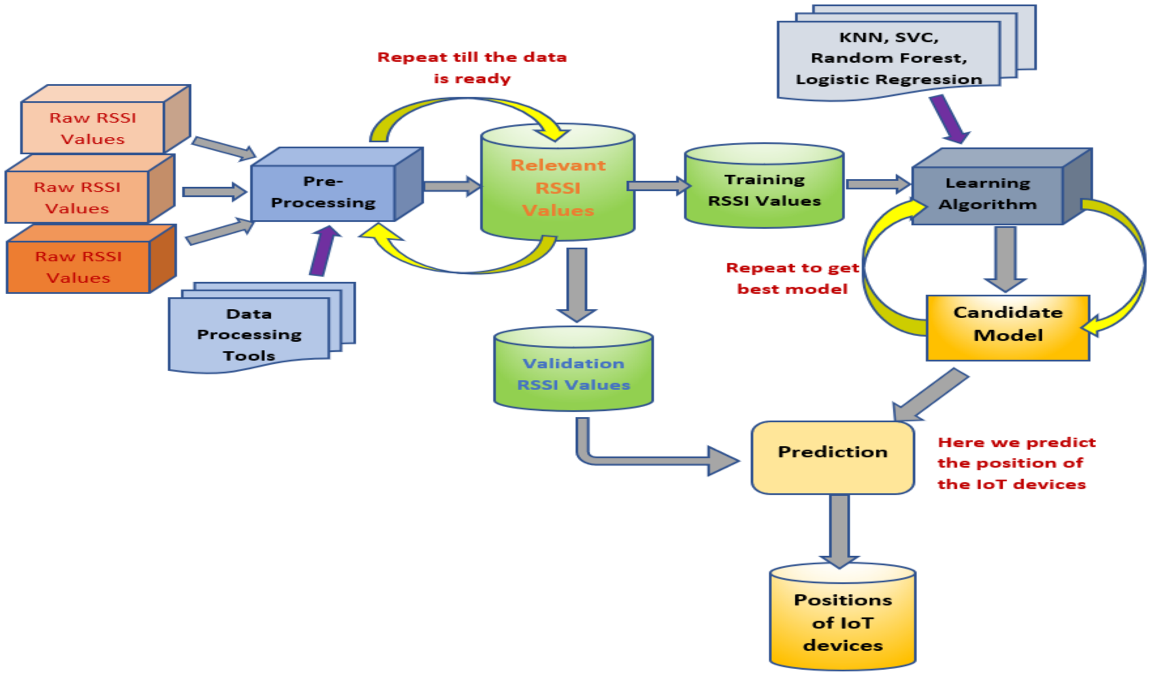

Figure 2: Workflow Diagram

Algorithm 1: Machine Intelligent Localization for Indoor Environment

Input: Sample dataset
Output: Target label

1. Start
2. IoT device broadcast beacon at time t;
2. Wn extracts the RSSI value;
3. Wn sends the RSSI value to C;
4. C receives n number of RSSI values for a IoT device;
5. These values are kept as a testing sample;
6. Target = Best Model(testing sample);
7. Target is sent to the IoT device from C;
8. End

3. 1 ML Models Used

The supervised machine learning models used for finding the best model is kNN classifier [22], RF classifier [23], SVM classifier [24], and voting classifier. These are discussed below.

3.1.1 K-NN Classifier

KNN algorithm is a simple Supervised ML algorithm technique. This algorithm collects all the available datasets that was processed and classified before and organises any new input on the basis of the similarity with previous databased. It is called as non-parametric algorithm because in KNN

there is no predefined mapping function. This algorithm is used to resolve classification and regression type problems. It is a lazy learner as it does not learn using training dataset it classifies the data when the prediction is required. At the training phase, the KNN algorithm stores the input data and when it gets new input, then it groups that input into a group that best fits newly entered data's group.
In this paper KNN is used to classify the exact location of IoT devices. i.e., to classify in which location the IoT device is present. K nearest neighbors is the best algorithm to store non-linear classified data points means when the data point is distributed in non-linear manner. Classification of new data point is happened by calculating the distance from the nearest neighbors' points means if K=3, then it should be found 3 distance from new data point. if the number of the distance from the is more from one type of point, then this new data point will

3.1.2 RF Algorithm

RF is a popular supervised ML algorithm technique. In this paper it is used for classification. RF uses the concept of ensemble learning; the main idea of this method is to combine the models to increase to the overall result. It means that this RF algorithm creates multiple decision trees on randomly selected from the data set, and it add them together to get the better accuracy and result. It is a supervised ML method which is used in both Regression and Classification problems. The RF takes the inputs from many decisions tree and based on the majority votes of the data the result is decided. Fig. 3 shows the implementation of RF model for localization of a device in indoor environment.

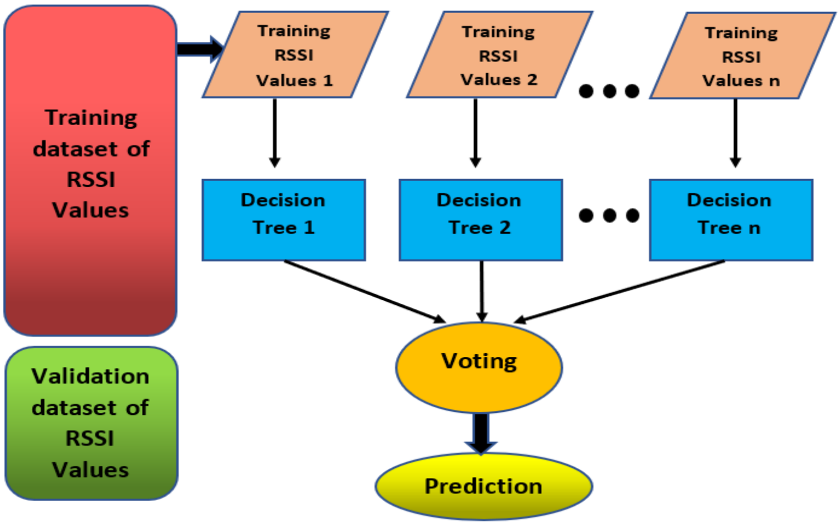

Figure 3: Random Forest Model

3.1.3 SVM Algorithm

SVM [24,28,29] is a ML algorithm which belongs to a class of classification algorithms. The original form of SVM was invented on the year 1963, but in the year 1992 a new way was invented to create a nonlinear classifier. These classes of algorithms are used for Classification algorithms are used to predict the binary values such as true or false, male or women, emails as spam or not, faulty, or not faulty in our case. More formally a SVM creates a hyper plane to separate linearly separable points and according to that it gives the prediction result. SVM is a supervise ML method which is used both classification and regression problems. SVM creates a hyperplane in multidimensional space to divide different classes. SVM finds a maximum hyper plane or line that segregates the dataset into classes.

3.1.4 Voting Classifier

Voting Classifier is a ML ensemble model that is formed by combination of more than one technique to predict the output with high value of functional parameter (accuracy, f1-call, precision, recall etc). It is used to add more than model by user and these models are supplied to voting classifier to and it gives high value of functional parameter. Rather than using different model, it is used as single model which is taken different model as input and gave output with high value accuracy. Fig. 4 shows the implementation of voting classifier.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

Voting Classifier are two types different types:
1. Hard Voting: This is a type of voting classifier which is based on majority voting.in mathematical term it is known as mode. There will be different output by different model, it is used to predict the class label by using the concept of mode. this technique mostly used in deep learning.
2. Soft Voting: In this type of classifier, the output class is predicted by taking the probability of all the output which is given by all the model then it finds the mean of that probability. The output which has most mean probability. that will be our final output. For example, let us assume that the probability for class A = (0.30, 0.47, 0.53) and B = (0.30, 0.47, 0.53) and C = (0.30, 0.47, 0.53) and D = (0.30, 0.47, (0.20, 0.32, 0.40). So, with an average of 0.4333 for class A and 0.3067 for class B, here class A has higher mean probability so the predicted class is A.
Hard Voting classifier formula [30]:

$$C(Y)=\text{mode}\{h_1(Y), h_2(Y), h_3(Y)\ldots\ldots\ldots\ldots\ldots\ldots\} \quad (1)$$

Soft Voting classifier formula [30]:

$$C(Y)=\arg\max\sum_{j=1}^{B} WjPij \quad (2)$$

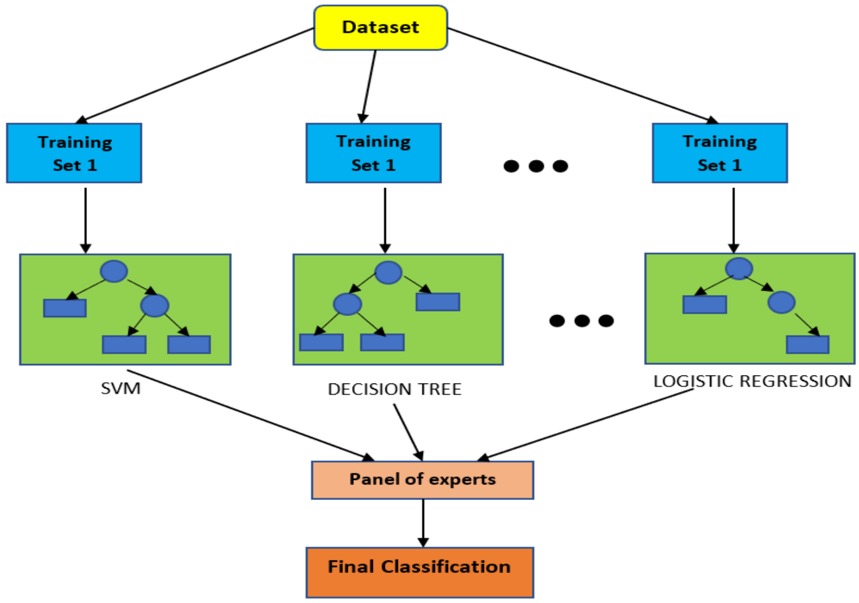

Figure 4: Voting classifier Model

Algorithm 2: Voting classifier

Input: Sample dataset
Output: Best Model
1. In training data set apply 3 classifiers (Random Forest, SVM, KNN);
2. Compare the performance parameter of 3 classifier;
3. Choose Either hard voting classifier or soft voting classifier for finding majority value;
4. Compare the performance parameter of voting classifier with model used (KNN, SVM, Random Forest);

3.2 Dataset Used
In this paper the dataset from [21] is used. The dataset is collected to analyse how Wi-Fi signal strengths (RSSI values) can be used for localization of an IoT device in an indoor setting. Characteristics of this data set is multivariant. Number of the instances used in this data set is 2000 and number of the attributes used in this dataset is 7. In this data set the RSSI values are received from various Wi-Fi beacons in a fixed location. Further, the dataset considers a setup of a large floor of a building. This floor has seven Wi-Fi beacons RSSI values that are received from these beacons are used to predict the position of different IoT devices in four different rooms, which are named as Room1, Room2, Room3 or Room4 respectively.

4. Performance Evaluation

The purposed work Localization of IOT devices has been done using the Python and the tools and libraries are used like jupyter notebooks, NumPy, SkLearn, this is because of the simplicity and modularity of the Python programming language and the modularity and functionality it produces with its numerous toolkits, function, libraries and built-in-classes for data processing, visualization, and implementation. The work has been carried out in the following manner

This paper work was carried on four different systems. The specifications of systems are as follows:

System 1: Operating system: windows 10 Home Single Language 64-bit, System Manufacturer: LENOVO, System Model: 81NG BIOS: CNCN19WW, Processor: Intel(R) Core (TM) i5-10210U CPU @1.60GHz (8 CPUs), Memory: 8192MB RAM, Name: Intel(R) UHD Graphics, DAC Type: Internal, Approx. Total Memory: 4138MB, Display Memory: 128MB, Shared Memory: 4010MB, Name: NVIDIA GeForce MX250, DAC Type : Integrated RAMDAC, Approx. Total Memory : 5993MB,Display Memory : 1983MB, Shared Memory : 4010MB.

System 2: Operating system: windows 10 Home Single Language 64-bit, System Manufacturer: ASUSTeK Computers INC., System Model : Strix 15 GL503GE, BIOS : GL503GE.316, Processor : Intel(R) Core(TM) i5-8300H CPU @2.30GHz(8 CPUs), Memory : 8192MB RAM, Name : Intel(R) UHD Graphics 630, DAC Type : Internal, Approx. Total Memory : 4151MB, Display Memory : 128MB, Shared Memory : 4023 MB, Name : NVIDIA GeForce GTX 1050 Ti, DAC Type : Integrated RAMDAC, Approx. Total Memory : 8044MB, Display Memory : 4021MB, Shared Memory : 4023MB.

System 3: Operating system: windows 10 Home Single Language 64-bit, System Manufacturer: HP, System Model: HP Pavilion Gaming Laptop 15-ec2xxx, BIOS: F.15, Processor: AMD Ryzen 5 5600H with Radeon Graphics (12 CPUs), Memory: 8192MB RAM

Name: AMD Radeon (TM) Graphics, DAC Type: Internal DAC (400MHz), Approx. Total Memory: 4258MB, Display Memory: 496MB, Shared Memory: 3762M, Name: NVIDIA GeForce GTX 1650, DAC Type: Integrated RAMDAC, Approx. Total Memory: 7724MB, Display Memory: 3962MB, Shared Memory: 3762MB.

System 4: Operating system: windows 10 Home Single Language 64-bit, System Manufacturer: HP, System Model: HP Pavilion Notebook, BIOS: F.74, Processor: Intel(R) Core (TM) i5-6200U CPU @2.30GHz (4 CPUs), Memory: 8192MB RAM, Name: Intel(R) HD Graphics 520, DAC Type: Internal, Approx. Total Memory: 4143MB Display Memory: 128MB, Shared Memory: 4015MB, Name: NVIDIA GeForce 940M, DAC Type: Integrated RAMDAC, Approx. Total Memory: 6025MB, Display Memory: 2010MB, Shared Memory: 4015MB.

In this work, firstly the data set is converted into .csv file then it is implemented as the input. The input is then pre-processed by slicing it into independent and dependent features and then scaled using the Sklearn's Standardscaler. By using supervised machine learning algorithm like SVM, KNN, Random Forest it classified the different the class from the feature value. Voting classifier is used as ensemble technique combining the methods like kNN, Random Forest and SVM to get improvements in the results. The accomplishment of the models is then checked by using performance metrices like accuracy, recall, precision, f1-score. The results are then verified using cross validation and checking the results for various weights for training and testing.

Here the test dataset that was taken from the UCI website [21] which was used in computation of the results. And then the assessment of our approach has been done. The python code has been cross validated 10 times and their average are taken for consideration. Classification methods of SVM, KNN, Random Forest and Voting Classifier has been used for this work. For a classification problem the following outcomes are possible:

1. True positives: the prediction belons to the class that it really belongs to.
2. True negatives: prediction does not belong to the class that it really belons to'
3. False positives: prediction belong to a class that it does not belong to
4. False negatives: prediction does not belong to a class when it should.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

The outcomes can be visualised in a confusion matrix. The main metrices for the evaluation of a classification model can also be called accuracy precision and recall.
Accuracy: It is the percentage of correct predictions for the test data.

$$\text{Accuracy} = \frac{Correct\ predictions}{all\ predictions} \tag{3}$$

Precision: It is defined as the ratio to the true positives to all which were predicted to belong in a certain class

$$\text{Precision}=\frac{true\ positives}{true\ positivesv+false\ positives} \tag{4}$$

Recall: It is defined ratio of the predictions that belong to a classification to the ones that really belong in that classification.

$$\text{recall} = \frac{true\ positives}{true\ positives+false\ negatives} \tag{5}$$

In table 1 it is shown that our data set is splitted into 60% training and 40% testing part.in table 1 there are different parameter values for the different model we used including ensemble technique (Voting classifier). The best performance of different functional parameter is given by KNN (accuracy-0.98, precision-0.99, f1score-0.99, recall-0.99) and voting classifier (accuracy-0.98, precision-0.99, f1score-0.99, recall-0.99).

4.1 Results and Discussion

In the below Fig. 5 it is shown the performance of different parameters of different models with different percentages. Each model is trained in 60% training and 40% testing dataset. Confusion-matrix is a matrix which can be used for assessing the behaviour and understanding the effectiveness of a binary or a multilevel classifier. all the confusion-matrix diagrams are made by heatmap. In Fig. 6 it is the confusion-matrix diagram made for the KNN. It shows the plotting between Y_Test and kNN prediction. In Fig. 7 it is the confusion-matrix diagram made for the SVM. It shows the plotting between Y_Test and SVM prediction. In Fig. 8 it is the confusion-matrix diagram made for the Random Forest. It shows the plotting between Y_Test and Random Forest prediction. In Fig. 9 it is the confusion-matrix diagram made for the Voting classifier. It shows the plotting between Y_test and voting classifier prediction.

Table 1: Performance analysis for Train set: 60% & Test set: 40%

ML Technique	Accuracy	Precision	F1 Score	Recall
KNN	0.98	0.99	0.99	0.99
SVM	0.97	0.98	0.98	0.98
Random Forest	0.98	0.98	0.98	0.98
Voting Classifier	0.98	0.99	0.99	0.99

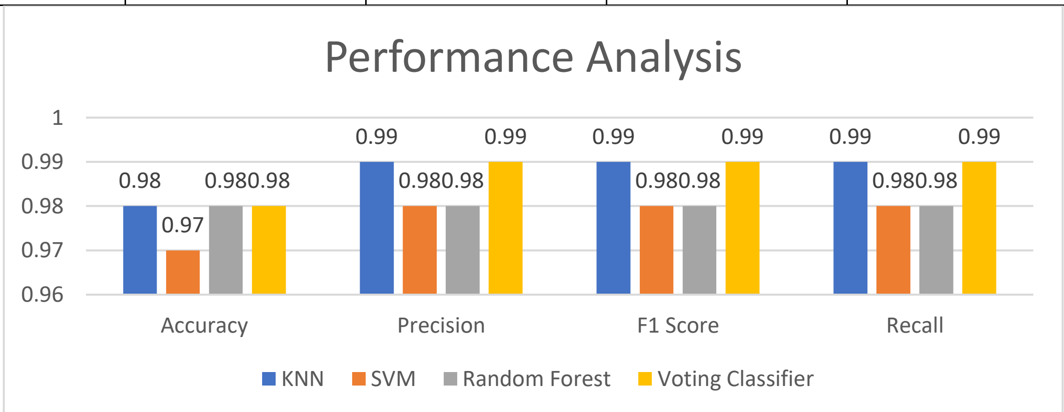

Figure 5: Performance analysis Train set: 60% and Test set: 40

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

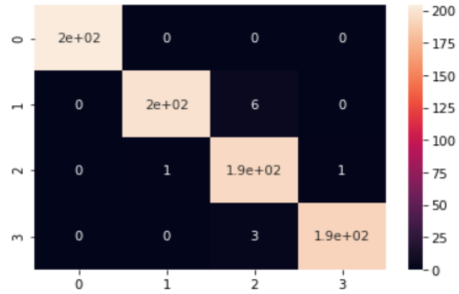

Figure 3: KNN Confusion Matrix for Train set 60% & Test set 40%

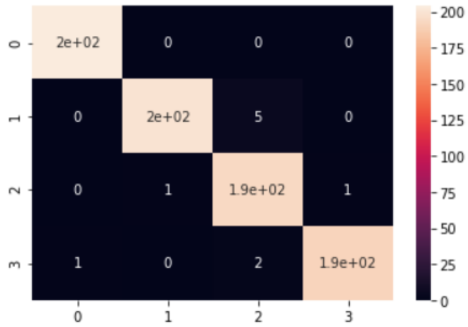

Figure 4: SVM Confusion Matrix for Train set 60% & Test set 40%

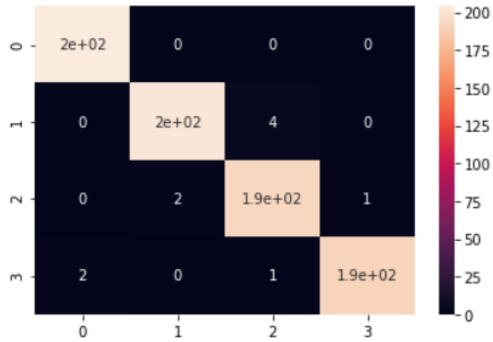

Figure 5 Random Forest Confusion Matrix for Train set 60% & Test set 40%

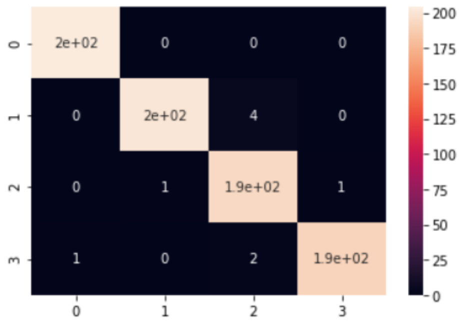

Figure 6: Voting Classification Confusion Matrix for Train set 60% & Test set 40%

In table 2 it is shown that our data set is splitted into 70% training and 30% testing part. In table 2 there are different parameter values for the different model we used including ensemble technique (Voting classifier). The best performance of different functional parameter is given KNN (accuracy-0.98, precision-0.98, f1score-0.98, recall-0.98) and voting classifier (accuracy-0.98, precision-0.98, f1score-0.98, recall-0.98). In the below Fig. 10 it is shown the performance of different parameters of different models with different percentages. Each model is trained in the ratio of Training dataset: Testing dataset of 70:30. In Fig. 11 it is the confusion-matrix diagram made for the KNN. It shows the plotting between Y_test and KNN prediction. In Fig. 12 it is the confusion-matrix diagram made for the SVM, it shows the plotting between Y_test and SVM prediction. In Fig. 13 it is the confusion-matrix diagram

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

made for the Random Forest, it shows the plotting between Y_test and Random Forest prediction. In Fig. 14 it is the confusion-matrix diagram made for the Voting classifier, it shows the plotting between Y_test and voting classifier prediction.

Table 2: Performance analysis for Train set: 70% & Test set: 30%

ML Technique	Accuracy	Precision	F1 Score	Recall
KNN	0.98	0.98	0.98	0.98
SVM	0.97	0.98	0.98	0.98
RF	0.98	0.98	0.98	0.98
Voting Classifier	0.98	0.98	0.98	0.98

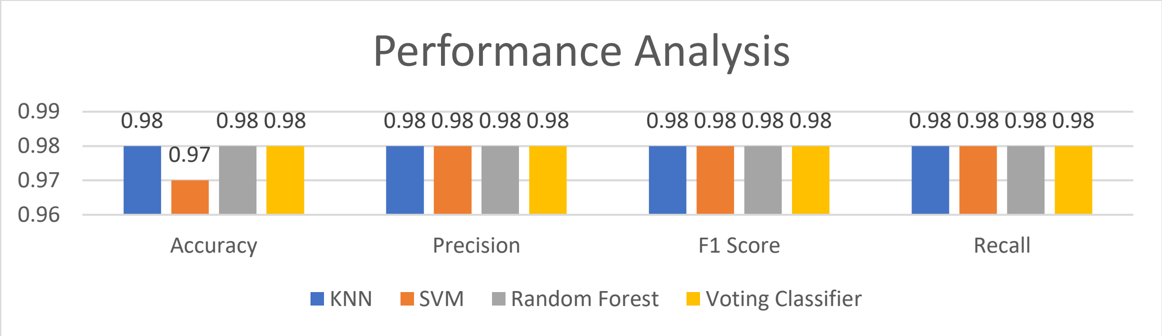

Figure 10: Performance analysis for Test set: 70% and Train set: 30%

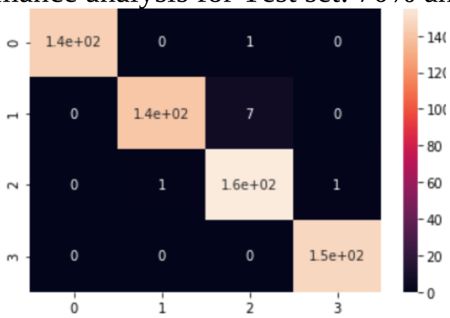

Figure 11: KNN Confusion Matrix for Train set 70% & Test set 30%

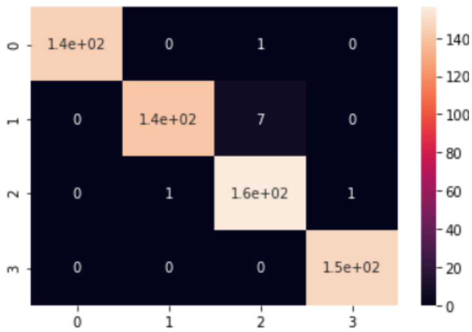

Figure 12: SVM Confusion Matrix for Train set 70% & Test set 30%

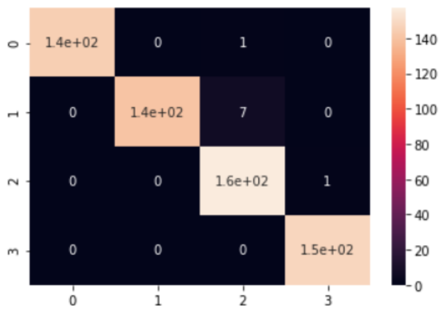

Figure 13: Fandom Forest Confusion Matrix for Train set 70% & Test set 30%

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

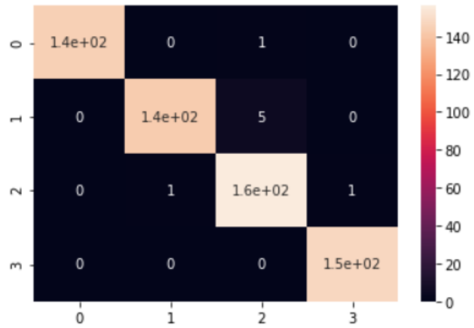
Figure 14: Voting Classifier Confusion Matrix for Train set 70% & Test set 30%

In table 3 it is shown that our data set is splitted into 80% training and 20% testing part. In table 3 there are different parameter values for the different model we used including ensemble technique (Voting classifier). The best performance of different functional parameter is given by KNN (accuracy-0.98, precision-0.99, f1score-0.99, recall-0.99) and voting classifier (accuracy-0.98, precision-0.99, f1score-0.99, recall-0.99). In the below Fig. 15, it is shown the performance of different parameters of different models with different percentages. Each model is trained in 80% training and 20% testing dataset. In Fig. 16 it is the confusion-matrix diagram made for the KNN, it shows the plotting between Y_test and KNN prediction. In Fig. 17 it is the confusion-matrix diagram made for the SVM, it shows the plotting between Y_test and SVM prediction. In Fig. 18 it is the confusion-matrix diagram made for the Random Forest, it shows the plotting between Y_test and Random Forest prediction. In Fig. 19 it is the confusion-matrix diagram made for the Voting classifier, it shows the plotting between Y_test and voting classifier prediction.

Table 3: Performance analysis for Train set: 80% & Test set: 20%

ML Technique	Accuracy	Precision	F1 Score	Recall
KNN	0.98	0.99	0.99	0.99
SVM	0.97	0.99	0.98	0.99
RF	0.98	0.99	0.99	0.99
Voting Classifier	0.98	0.99	0.99	0.99

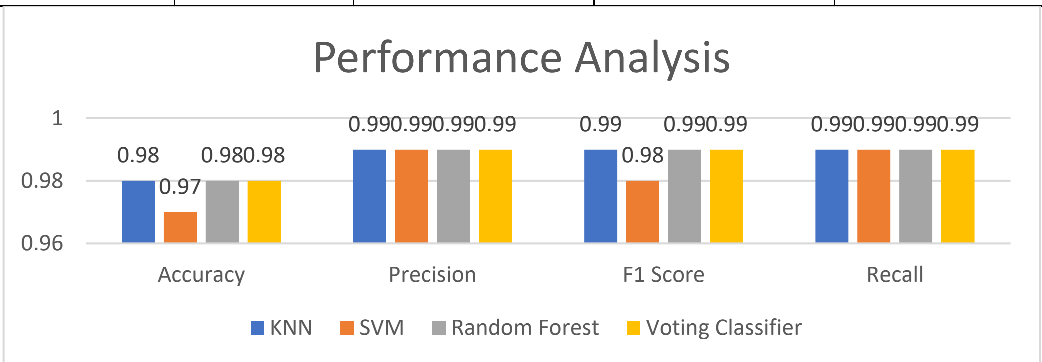

Figure 15: Performance analysis for Train set: 80% & Test set: 20%

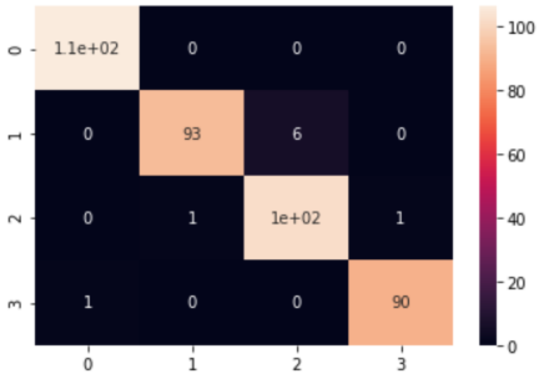
Figure 16: KNN Confusion Matrix for Train set: 80% & Test set: 20%

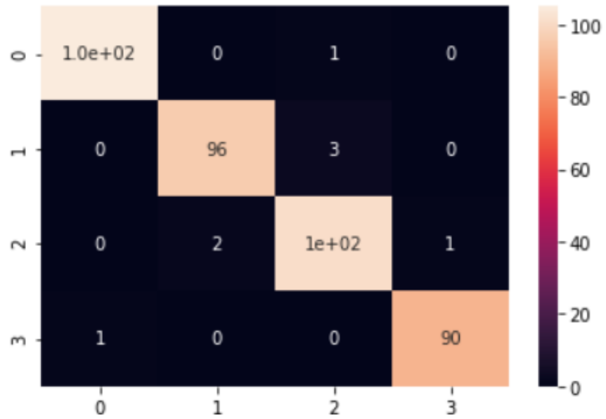

Figure 17: SVM Confusion Matrix for Train set: 80% & Test set: 20%

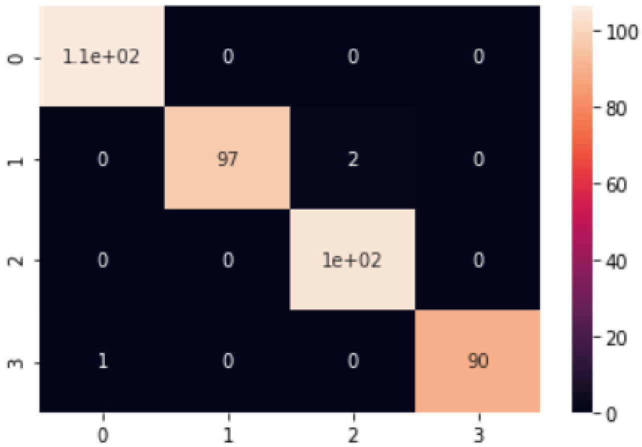

Figure 18: Random Forest Confusion Matrix for Train set: 80% & Test set: 20%

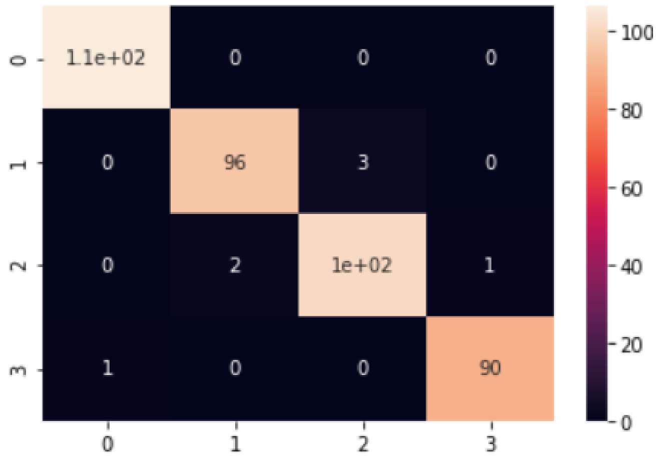

Figure19: Voting Classifier Confusion Matrix for Train set: 80% & Test set: 20%

In table 4 it is shown that our data set is splitted into 75% training and 25% testing part. In table 4 there are different parameter values for the different model we used including ensemble technique (Voting classifier). The best performance of different functional parameter is given KNN (accuracy-0.98, precision-0.98, f1score-0.98, recall-0.98) and voting classifier (accuracy-0.98, precision-0.98, f1score-0.98, recall-0.98). In the below Fig. 20 it is shown the performance of different parameters of different models with different percentages. Each model is trained in 75% training and 25% testing dataset. In Fig. 21 it is the confusion-matrix diagram made for the KNN, it shows the plotting between Y_test and KNN prediction. In Fig. 22 it is the confusion-matrix diagram made for the SVM, it shows the plotting between Y_test and SVM prediction. In Fig. 23 it is the confusion-matrix diagram made for the Random Forest, it shows the plotting between Y_test and Random Forest prediction. In Fig. 24 it is the confusion-matrix diagram made for the Voting classifier, it shows the plotting between Y_test and voting classifier prediction.

Table 4: Performance analysis for Train set: 75% & Test set: 25%

ML Technique	Accuracy	Precision	F1 Score	Recall
KNN	0.98	0.98	0.98	0.98
SVM	0.97	0.99	0.99	0.99
RF	0.98	0.98	0.98	0.98
Voting Classifier	0.98	0.98	0.98	0.98

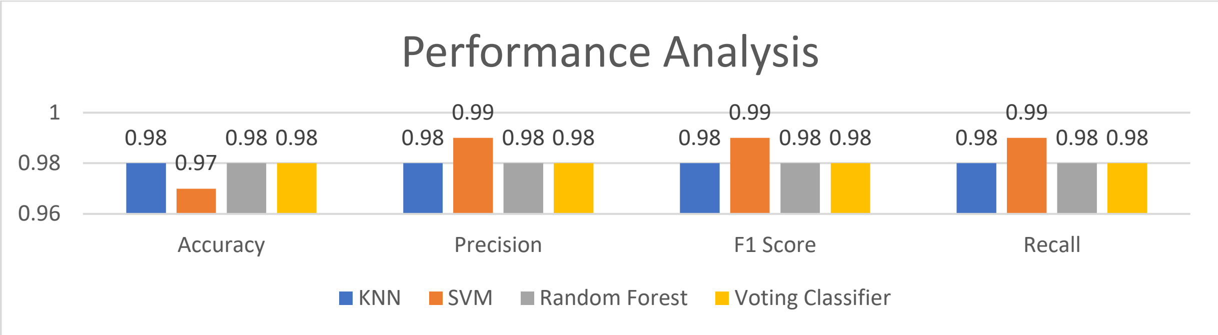

Figure 20: Performance analysis for Train set: 75% & Test set: 25%

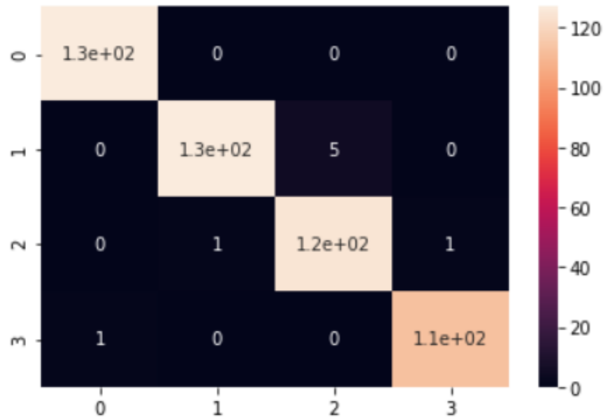

Figure 21: KNN Confusion Matrix for Train set 75% & Test set 25%

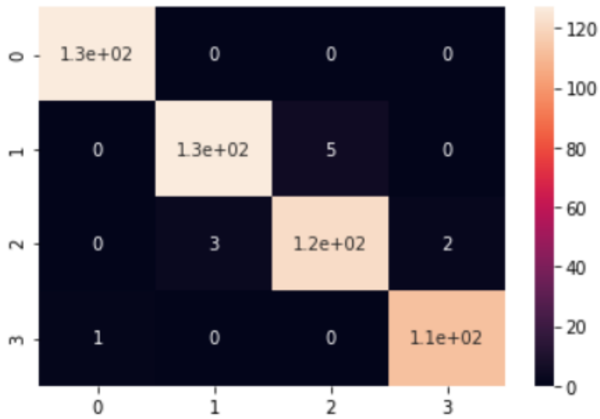

Figure 22: SVM Confusion Matrix for Train set 75% & Test set 25%

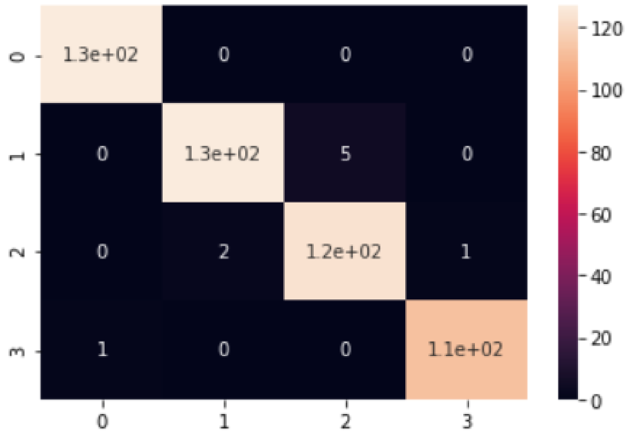

Figure 23: Random Forest Confusion Matrix for Train set 75% & Test set 25%

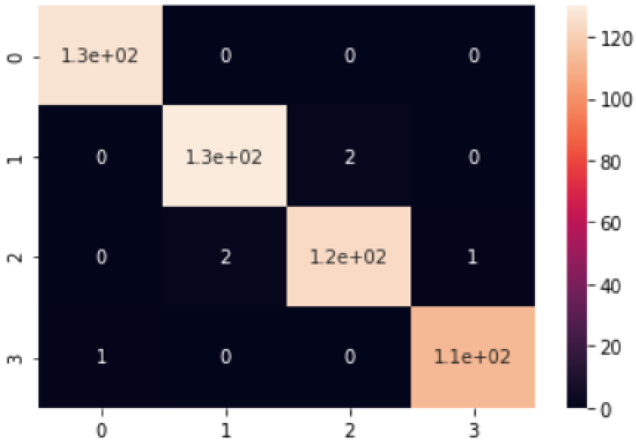

Figure 24: Voting Classifier Confusion Matrix for Train set 75% & Test set 25%

4.2 Hardware Experimentation

We integrate this algorithm into the location-based services reference model as mentioned before. We use this approach to locate the IOT device using Wi-Fi signal strength of different mobile application. First it has been considered mobile as IOT devices. Then we took 3 mobile devices and installed WI-FI Router Master [32] app in it. WI-FI Router Master app is used to detect the Wi-Fi signal strength of different device and speed of the Wi-Fi signal. First, we placed the server 1 at class 1 and placed 3 devices in class 1 in different corners and connected the 3 devices to the hotspot of the server1. After that we took the reading of RSS values i.e., (received signal strength). From D-1 the values ranged between -45 to -54 and from D-2 the values ranged in between -51 to -67. For D-3 the value was ranged in between -60 to -68. In this way we took the reading in class 1 and inserted the RSS values under D-1 (Device 1), D-2 (Device 2), and D-3 (Device 3) columns. Similarly, we turned on the hotspot of the server 2 in class 2 and connected it to the three devices named D-1, D-1, D-3 placed in class 1. As we did before we took the RSS values which Ranged in between -67 to -78 for D-1, -59 to -65 for D-2 and -68 to -76 for D-3. Then, we inserted found RSS values for server-2 in the database. In this way we created our database. Our database contains four columns named as D-1, D-2, D-3, and Class. There are 50 RSS values for each device. Both class 1 and class 2 contain 25 values each. Each csv files have been merged and resulted in a single csv file which includes both healthy and non-healthy class. After that the dataset has been used in different machine learning algorithm in which KNN, SVM, Random Forest and voting classifier has been used. In this experiment, all the models and voting classifier gives 100% of accuracy, precision, F1 score and Recall. The experimental setup is shown in Fig. 25. The results of this experiment are shown in Table 5 and figures 26-31. The results clearly represent the performance parameters values with accuracy of 100%, and also confusion matrix of different models.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

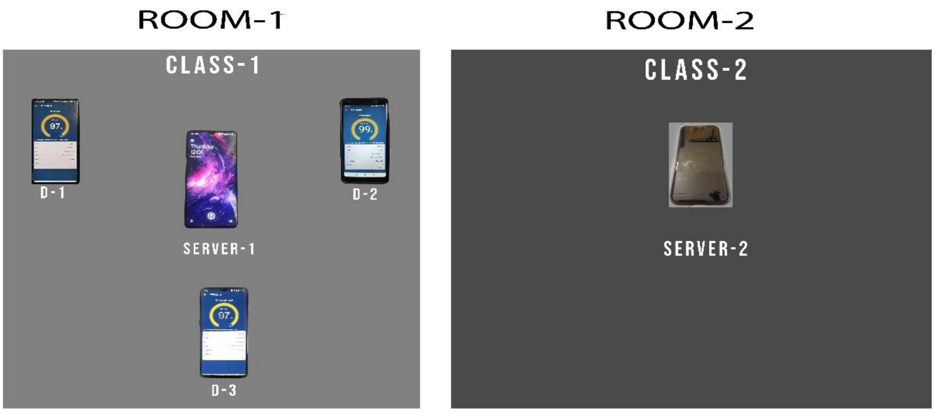

Figure 25: Experimental Set-up

	D-1	D-2	D-3	CLASS
0	-52	-67	-67	1
1	-51	-64	-64	1
2	-54	-59	-60	1
3	-48	-56	-62	1
4	-46	-62	-64	1
5	-49	-54	-62	1
6	-51	-53	-66	1
7	-47	-56	-64	1
8	-48	-52	-68	1
9	-50	-53	-65	1

Figure 26: Snapshot of Experimental Dataset

Table 5: Performance analysis for Test set:75% Train set: 25% of the experimental Dataset

ML Technique	Accuracy	Precision	F1 Score	Recall
KNN	1.00	1.00	1.00	1.00
SVM	1.00	1.00	1.00	1.00
RF	1.00	1.00	1.00	1.00
Voting Classifier	1.00	1.00	1.00	1.00

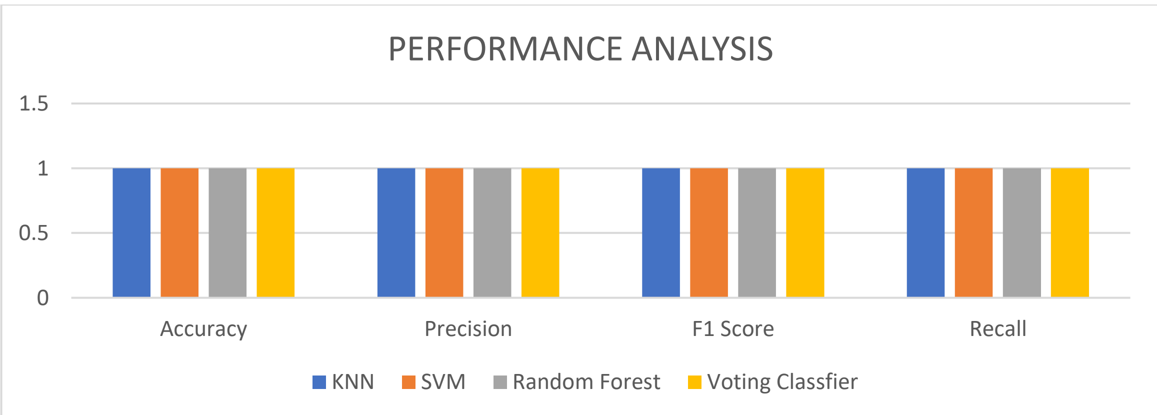

Figure 27: Performance analysis for Test set:75% Train set: 25% for the Experimental Dataset

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

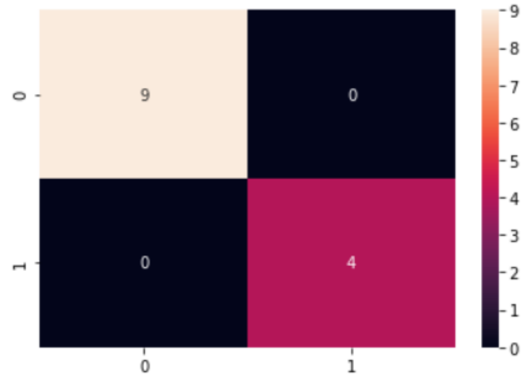

Figure 28: KNN-Confusion matrix

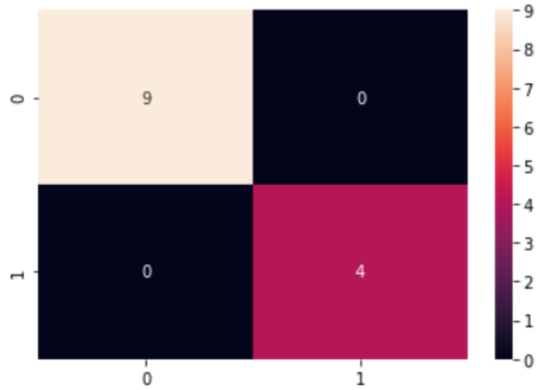

Figure 29: Random Forest-Confusion matrix

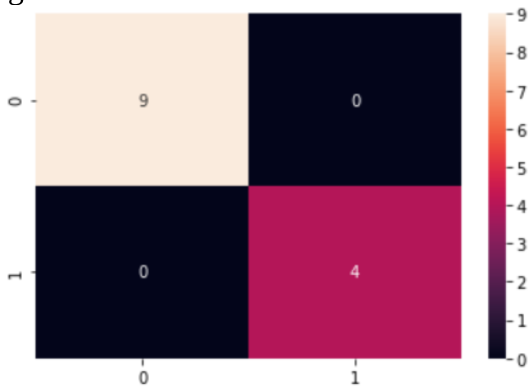

Figure 30: SVM-Confusion matrix

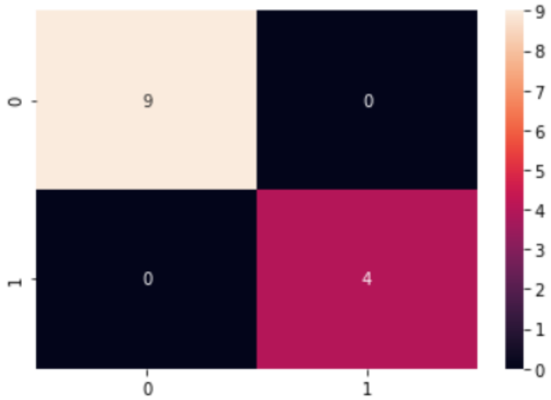

Figure 31: Voting Classifier-Confusion matrix

6. Conclusion

In this work, we have proposed this localization technique because in indoor environments like schools, colleges, libraries, theatres etc. there is no direct sight between user and the satellite as a result of which the positioning system of maps does not work properly. There is no error free technique/method for the localization of IoT devices and every new introduced technology and approach must be thoroughly evaluated. Here we have used different types of supervised ML algorithms, in order to improve the localization and position estimation. Here we introduced a technique for indoor positioning of different IoT devices networks by reference from recent research and work in this field. In this work we use kNN, SVM, RF and the ensemble technique Voting Classifier algorithm for the hybridization and were able to record accuracy, f1 score, recall and precision of above 99%. A real-world hardware experiment for localization of IoT devices has been done which had attained an accuracy of 100%. Till date the most popular method of positioning of anything is done by GPS. But for indoor environments where satellite signals are very low, we require a new and robust solution. So, this area needs to be explored more for enhancing the location accuracy and reducing the positioning error.

References

[1] Ma, S., Liu, Q., & Sheu, P. C. Y. (2017). Foglight: Visible light-enabled indoor localization system for low-power IoT devices. *IEEE Internet of Things Journal*, *5*(1), 175-185.

[2] Röhrig, C., & Müller, M. (2009, October). Indoor location tracking in non-line-of-sight environments using a IEEE 802.15. 4a wireless network. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 552-557). IEEE.

[3] Bargh, M. S., & de Groote, R. (2008, September). Indoor localization based on response rate of bluetooth inquiries. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments* (pp. 49-54).

[4] He, S., & Chan, S. H. G. (2015). Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, *18*(1), 466-490.

[5] Moghtadaiee, V., & Dempster, A. G. (2014). Indoor location fingerprinting using FM radio signals. *IEEE Transactions on Broadcasting*, *60*(2), 336-346.

[6] Shala, U., & Rodriguez, A. (2011). Indoor positioning using sensor-fusion in android devices.

[7] Otsason, V., Varshavsky, A., LaMarca, A., & De Lara, E. (2005, September). Accurate GSM indoor localization. In *International conference on ubiquitous computing* (pp. 141-158). Springer, Berlin, Heidelberg.

[8] Liu, H., Yang, J., Sidhom, S., Wang, Y., Chen, Y., & Ye, F. (2013). Accurate WiFi based localization for smartphones using peer assistance. *IEEE Transactions on Mobile Computing*, *13*(10), 2199-2214.

[9] Jeong, J. P., Yeon, S., Kim, T., Lee, H., Kim, S. M., & Kim, S. C. (2018). SALA: Smartphone-assisted localization algorithm for positioning indoor IoT devices. *Wireless Networks*, *24*(1), 27-47.

[10] Bregar, K., & Mohorčič, M. (2018). Improving indoor localization using convolutional neural networks on computationally restricted devices. *IEEE Access*, *6*, 17429-17441.

[11] Bianchi, V., Ciampolini, P., & De Munari, I. (2018). RSSI-based indoor localization and identification for ZigBee wireless sensor networks in smart homes. *IEEE Transactions on Instrumentation and Measurement*, *68*(2), 566-575.

[12] Ouameur, M. A., Caza-Szoka, M., & Massicotte, D. (2020). Machine learning enabled tools and methods for indoor localization using low power wireless network. *Internet of Things*, 100300.

[13] Zhang, D., Yang, L. T., Chen, M., Zhao, S., Guo, M., & Zhang, Y. (2014). Real-time locating systems using active RFID for Internet of Things. *IEEE Systems Journal*, *10*(3), 1226-1235.

[14] Sadowski, S., & Spachos, P. (2018). Rssi-based indoor localization with the internet of things. *IEEE Access*, *6*, 30149-30161.

[15] Cai, C., Hu, M., Cao, D., Ma, X., Li, Q., & Liu, J. (2019). Self-deployable indoor localization with acoustic-enabled IoT devices exploiting participatory sensing. *IEEE Internet of Things Journal*, *6*(3), 5297-5311.

[16] Bhatti, M. A., Riaz, R., Rizvi, S. S., Shokat, S., Riaz, F., & Kwon, S. J. (2020). Outlier detection in indoor localization and Internet of Things (IoT) using machine learning. *Journal of Communications and Networks*, *22*(3), 236-243.
[17] Le, M. T. (2021). Enhanced Indoor Localization Based BLE Using Gaussian Process Regression and Improved Weighted kNN. *IEEE Access*, *9*, 143795-143806.
[18] Singh, A., Kotiyal, V., Sharma, S., Nagar, J., & Lee, C. C. (2020). A machine learning approach to predict the average localization error with applications to wireless sensor networks. *IEEE Access*, *8*, 208253-208263.
[19] Huang, C. H., Lee, L. H., Ho, C. C., Wu, L. L., & Lai, Z. H. (2014). Real-time RFID indoor positioning system based on Kalman-filter drift removal and Heron-bilateration location estimation. *IEEE Transactions on Instrumentation and Measurement*, *64*(3), 728-739.
[20] Tran, H. Q., & Ha, C. (2019). Fingerprint-based indoor positioning system using visible light communication—a novel method for multipath reflections. *Electronics*, *8*(1), 63.
[21] Retrieved June 20, 2022, from https://archive.ics.uci.edu/ml/datasets/Wireless+Indoor+Localization
[22] Retrieved June 20, 2022, from https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification
[23] Retrieved June 20, 2022, from https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees
[24] Retrieved June 20, 2022, from https://scikit-learn.org/stable/modules/svm.html#classification
[25] Retrieved June 20, 2022, from https://dataaspirant.com/k-nearest-neighbor-classifier-intro/
[26] Retrieved June 20, 2022, from https://www.datacamp.com/community/tutorials/random-forests-classifier-python
[27] Retrieved June 20, 2022, from http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf
[28] Retrieved June 20, 2022, from https://medium.com/nerd-for-tech/support-vector-machine-92fa3c57d33b
[29] Retrieved June 20, 2022, from https://www.researchgate.net/figure/Pseudocode-for-SVM-Classifier_fig1_331875695
[30] Retrieved June 20, 2022, from https://web.stanford.edu/~hastie/THESES/gareth_james.pdf
[31] Retrieved June 20, 2020, from https://www.acadpubl.eu/hub/2018-118-24/1/6.pdf
[32] Retrieved June 20, 2022, from https://play.google.com/store/apps/details?id=com.wifianalyzer.speedtest.wifirouter.wifibooster
[33] Perković, T., Dujić Rodić, L., Šabić, J., & Šolić, P. (2023). Machine Learning Approach towards LoRaWAN Indoor Localization. *Electronics*, *12*(2), 457.
[34] Yadav, P., & Sharma, S. C. (2023). A systematic review of localization in WSN: Machine learning and optimization-based approaches. *International Journal of Communication Systems*, *36*(4), e5397.