5G ML Challenge

Classification of Home Network Users to Improve User Experience

TEAM - LAMDA

Srujana Talla

A20301022

Gowtham Bhupathiraju

A20348217

Naveen Kumar Rai

A20294516

Kodjo Opoku Botchway

A20338464

Table of Contents

Executive Summary

Internet access has recently become one of the most common household conveniences. The internet is connected to everything from the door lock to the cameras on each device. Internet service providers are interested in categorizing home network users to improve the user experience. The main objective is to design a model which will classify users into good category experience and bad user experiences. There are a lot of factors that impact the user's online experience. The DPI probe may be used to distinguish the end-to-end broadband network's uplink and downlink network sides. The data we were provided with contained eight key indicators. The indicators are obtained from the three-way handshake process, the data transmission process after the handshake is successful, and the numerical value of the indicator in milliseconds. We will classify users from the eight indicators' data as Users with Bad Experiences(UBE) or Users with Good Experiences(UGE).

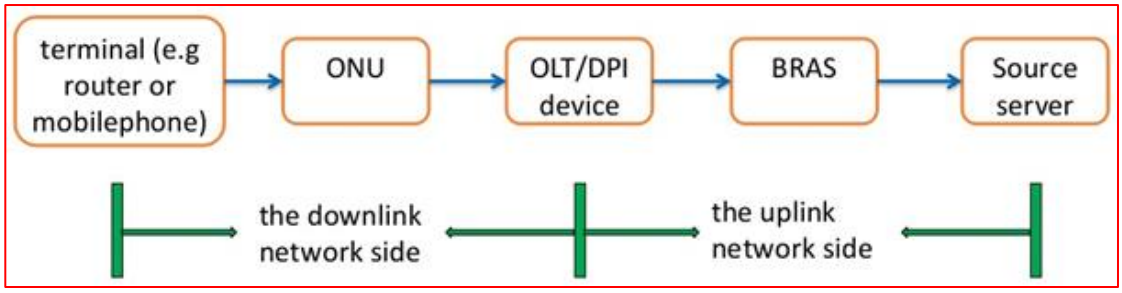
Raw data is parsed, cleaned, converted, and preprocessed before modeling and analysis. For the missing data, data is frequently imputed, a technique used to fill in or replace missing values that are conceptually similar to interpolation. We selected the potentially viable models based on data compatibility as part of investigating basic models. The data used to develop the model is non-representative, of poor quality, rife with errors, and so on. The quantity, quality, preparation, and data selection are all important factors for a better model solution. In the experimentation, Strategy-1, we performed min-max normalization for Random Forest Classifier, XGBoost, K Nearest Neighbors, and Support Vector Machine Classifier.

Furthermore, feature values are scaled and standardized (feature scaling). Feature scaling brings the value ranges of different features closer to help prevent certain features from dominating models and predictions and avoid computing problems. When the data used to create the model are not entirely representative of the scenarios for which the model may be utilized, selection bias develops, especially when working with new and unknown data. That brings in the Strategy-2 Multilayer Neural Network. Based on facts, Multilayer Neural Network is effectively the model that can achieve the solution goals for a given problem.

Business Understanding

An Internet connection enables us to surf the web, use IoT and smart devices, play games, watch movies and music, communicate with loved ones, keep up on work and education, and much more—all from the comfort of our homes. With the rapid development of mobile internet, home broadband has become an integral part of people's daily lives, making the market more and more saturated. User experience and broadband quality have become key factors in determining market competitiveness. Detecting network quality problems promptly and improving the user experience have attracted operators' attention.

A variety of things influence the online experience of the user. The uplink and downlink network sides can be separated using the DPI probe from the end-to-end broadband network. Figure 1 displays the network configuration. A significant portion of network issues is downlink network side issues. Classification of home network users to improve user experience is important as operators need to identify potentially unhappy users in advance. To do this, they need to properly differentiate between users with terrible experiences (hereafter, UBE) and users with excellent experiences (hence, UGE) by changes in downlink network side indicators.



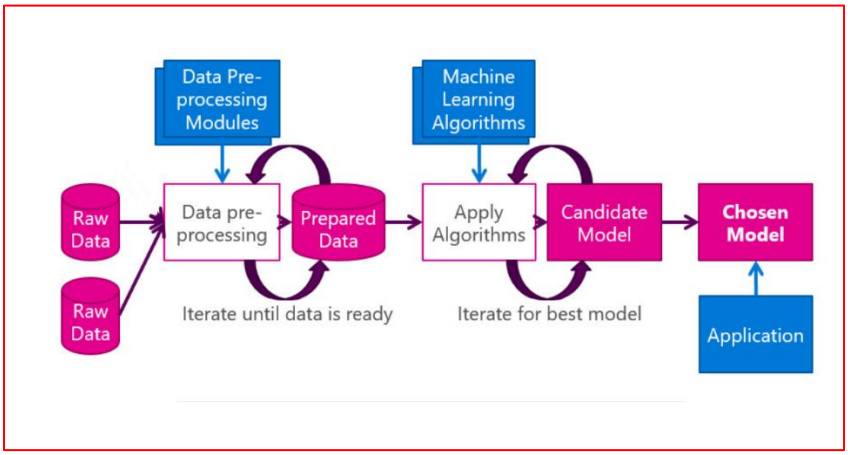
Data Understanding

There are more than 15 indicators, of which eight(8) network indicators are the key. We were provided eight(8) main key indicators from real networks. We measure the value of each indicator from the DPI device.

The data we were provided with contained eight key indicators listed below. The indicators are obtained from the three-way handshake process, the data transmission process after the handshake is successful, and the numerical value of the indicator in milliseconds. The specific physical meanings of the indicators are as follows:

1. **Indicator 1:** In the first step of the three-way handshake, the time interval between the syn ack packet and the ack packet;
2. **Indicator 2:** In the second step of the three-way handshake, the time interval between the syn ack packet and the ack packet;
3. **Indicator 3:** The time interval between the ack packet and the first payload packet in the three-way handshake;
4. **Indicator 4:** The response delay of the first packet with payload after the establishment of TCP for multiple flows in the session;
5. **Indicator 5:** In TCP transmission, the actual delay of transmission from the DPI position to the user terminal;
6. **Indicator 6:** In TCP transmission, the transmission delay from the DPI position to the website;
7. **Indicator 7:** In TCP transmission, the percentage of downlink retransmitted packets in the current session;
8. **Indicator 8:** In TCP transmission, the percentage of upstream retransmission packets of the current session.

Project Life Cycle :



Data Science is a synthesis of two fields data and science. Data may be anything actual or imagin ed, and science is the methodical study of the physical and natural worlds. So Data Science is not hing more than the systematic study of data and the derivation of knowledge via testable methodologies to make predictions about the Universe. Simply said, it is the application of science to data of any scale and from any source. Data has evolved into a new fuel that propels enterprises today. That is why it is critical to understand the life cycle of a data science project.

Applying algorithms for machine learning and statistical approaches that improve prediction models is part of the overall data science life cycle. Data extraction, preparation, cleaning, modeling, and assessment are the process's most typical data science stages.

Data Preparation

Data from each user was received in the form of CSV files. There were 50 customers with a good user experience and 50 with a terrible user experience in the validation and test group. We were given either UGE or UBE in quantities of 150. Over seven days (6/10/2021 - 6/16/2021), each user in the data gathered several timestamps.

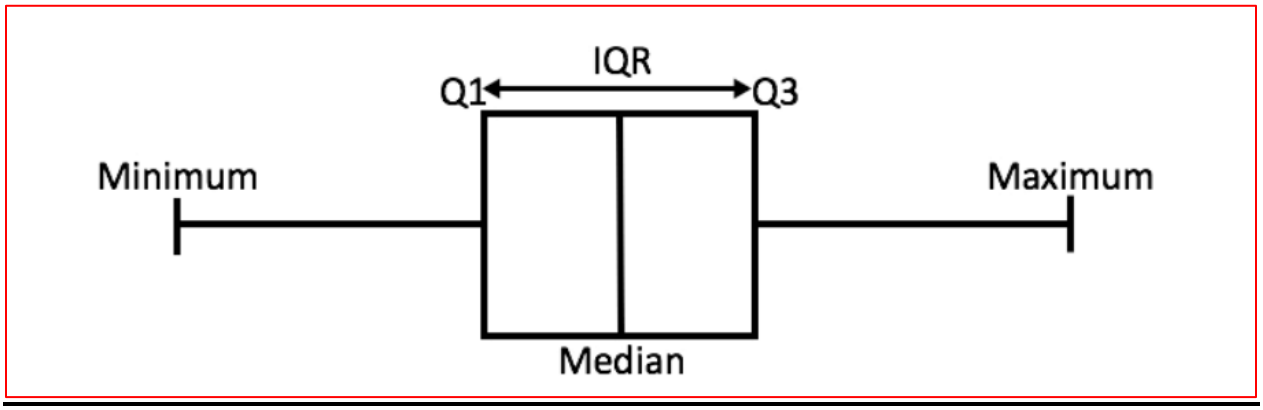
We combined the training users into one file, the test users into another, and the validation users into another. After that, we established a variable called "ID" to identify each user in the data we were given. The user experience type was added as a binary variable named "Type." We assigned values to the user's experience, starting with 0 for the UBE and ending with 1 for the UGE.

We made several separate files. In the first file, we aggregated the data by individual timestamps and calculated the mean of all entries for that period. We repeated the process, but this time on an hourly basis. Finally, we consolidated the data, creating a record with one row for each user. We attempted several types of aggregations to feed classification models with different versions of preprocessed data because the user data was relatively dense. We repeated the method but modified the aggregate to the median rather than the mean. Each of these files was created for test, training, and validation sets, for a total of 18 files.

Checking for duplicates in data is usually crucial; however, in our situation, we did not identify any duplicates and did not have to eliminate the data owing to this typical issue.

Outlier detection and Data cleaning:

We used IQR methods to detect the outliers and remove them from the data as part of data cleaning. To explain it further, let us take the below picture as a reference:



This box plot gives us an idea of the current distribution of raw data.

The 'minimum' in the above box plot represents the minimum value of our dataset.

The 'maximum' in the above box plot represents the maximum value of our dataset.

The difference between the maximum and minimum gives us the range of our dataset.

Q1 is the first quartile of data, which means 25 percent of the data lies between the minimum and Q1.

The 'median' in the box plot represents the data's center point or second quartile.

Q3 is the third quartile of data, which means 75 percent of the data lies between the minimum and Q3.

IQR, or Inter-Quartile Range, is the difference between Q3 and Q1.

To find the outlier using the IQR method, we defined a new range called the decision range, and any data point out of this range is considered an outlier.

Lower Bound = (Q1 - 1.5*IQR)

Upper Bound = (Q3 + 1.5*IQR)

The data points lower than the "Lower Bound" or higher than the "Upper Bound" were considered outliers, and we removed them from the data.

Data Merging:

The training data set provided had 150 good users and 150 bad users. Each user's activity from 10/06/2021 to 16/06/2021 which is 6 days of user activity, is recorded. We combined the data

given for each user into a master data set which consisted of 300 users who had users in common.

Users with good experiences were assigned one, and users with bad experiences were assigned a 0 in the data set for model creation.

Summary Statistics of the Indicators:

```
In [7]: df.describe()
```

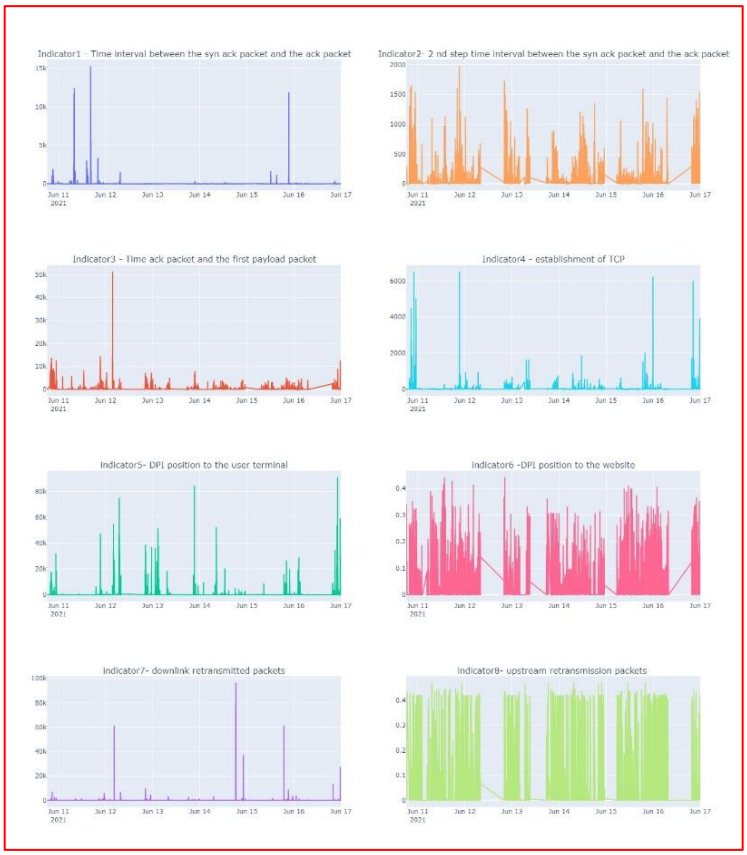
Out[7]:

	Unnamed: 0	hour	indicator1	indicator2	indicator3	indicator4	indicator5	indicator6	indicator7	indicator8	
count	3.687372e+06	3.687372e+06	3.687372e+06	3.687372e+06	3.687372e+06	3.687372e+06	3.687372e+06	3.687372e+06	3.687372e+06	3.687372e+06	3
mean	8.231628e+03	1.339846e+01	8.158119e+01	1.894195e+02	1.358366e+02	1.644803e+02	4.745937e+01	2.181715e+01	2.207362e-02	1.701789e-02	7
std	6.652201e+03	6.809274e+00	1.526898e+03	2.037408e+03	1.715116e+03	2.925861e+03	1.009032e+02	3.211521e+02	5.487697e-02	6.643991e-02	4
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1
25%	3.116000e+03	8.000000e+00	4.400000e+00	3.000000e+00	6.400000e-01	4.500000e+00	5.000000e+00	8.000000e+00	0.000000e+00	0.000000e+00	3
50%	6.678000e+03	1.400000e+01	1.179000e+01	6.890000e+00	2.580000e+00	1.275000e+01	1.900000e+01	1.500000e+01	0.000000e+00	0.000000e+00	7
75%	1.180100e+04	2.000000e+01	2.250000e+01	3.900000e+01	8.220000e+00	2.400000e+01	4.900000e+01	2.500000e+01	1.390000e-02	0.000000e+00	1
max	4.382700e+04	2.300000e+01	4.874515e+05	3.036850e+05	4.327250e+05	5.753230e+05	2.000000e+03	3.620020e+05	5.000000e-01	5.000000e-01	1

User – Good Experience:

Covariance Matrix of Indicators:

User – Bad Experience:



Covariance Matrix of Indicators:

Modeling

Strategy 1:

Data Preprocessing:

As a first strategy, data is grouped by User_ID, and specific time and mean of grouped values are calculated. Duplicate values are removed.

Data columns (total 23 columns):

#	Column	Non-Null Count	Dtype
0	day	1010 non-null	object
1	hour	1010 non-null	int64
2	specifictime	1010 non-null	object
3	indicator1	1010 non-null	float64
4	indicator2	1010 non-null	float64
5	indicator3	1010 non-null	float64
6	indicator4	1010 non-null	float64
7	indicator5	1010 non-null	float64
8	indicator6	1010 non-null	float64
9	indicator7	1010 non-null	float64
10	indicator8	1010 non-null	float64
11	User_ID	1010 non-null	object
12	Date	1010 non-null	object
13	Time	1010 non-null	object
14	indicator1_mean	1010 non-null	float64
15	indicator2_mean	1010 non-null	float64
16	indicator3_mean	1010 non-null	float64
17	indicator4_mean	1010 non-null	float64
18	indicator5_mean	1010 non-null	float64
19	indicator6_mean	1010 non-null	float64
20	indicator7_mean	1010 non-null	float64

```
21  indicator8_mean  1010 non-null   float64
22  True_Value       1010 non-null   int64
```

Min-Max Scaling:

We tried the min-max scaling method to deal with the outliers. As part of this, the scalar takes each value, subtracts the minimum, and divides by the range. The resultant values range between 0 and 1. The reason behind using this approach is to reduce the standard deviation to minimize the effect of outliers on the model's performance.

```
# Creating test/train datasets
X_train = dftrain.drop(['True_Value'], axis =1)
y_train = dftrain['True_Value']
X_test = dftest.drop(['True_Value'], axis = 1)
y_test = dftest['True_Value']
```

```
# Prefomring min-max normalization

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

scaler = MinMaxScaler()
model=scaler.fit(X_train)

x_train=model.fit_transform(X_train)
x_test=model.fit_transform(X_test)
```

Random Forest Classifier

Supervised machine learning methods, such as the random forest, are commonly used in regression and classification problems. It builds decision trees on diverse samples and utilizes their average for classification and popular vote for regression. One of the Random Forest Algorithm's most important characteristics is its capacity to handle data sets with both continuous variables, as in regression, and categorical variables, as in classification. It produces superior results when it comes to categorization challenges.

The Random Forest classification model's estimated model accuracy is 47.73%.

```
Scikit-Learn's Random Forest Classifier's prediction accuracy is: 47.73
Time consumed for training: 0.031 seconds
Time consumed for prediction: 0.00261 seconds
```

XGBoost

Extreme Gradient Boosting (XG Boost) is a package that focuses on computing speed and model accuracy. Gradient boosting, Stochastic Gradient Boosting, and Regularized Gradient Boosting are all supported.

The gradient-boosted tree is a popular method that is successfully implemented in open-source software called XGBoost. Gradient boosting is a supervised learning procedure that attempts to predict a target variable by combining the predictions of numerous weaker, simpler models.

The XG Boost model's estimated model correctness is 48.70%.

```
XGBoost's prediction accuracy is: 48.70
Time consumed for training: 0.149
Time consumed for prediction: 0.00588 seconds
```

K Nearest Neighbors

Based on the eight main signs, K-Nearest Neighbors (KNN) algorithm was employed to determine if a user was experiencing a good or terrible experience. This approach uses supervised machine learning. KNN calculates the distances between a query and each example in the data, selects the K instances closest to the inquiry, and then, in the classification phase, votes for or average the classifications (in the case of regression).

The K-Nearest Neighbors model's estimated model accuracy is 46.10%.

```
Scikit-Learn's K Nearest Neighbors Classifier's prediction accuracy is: 46.10
Time consumed for training: 0.003 seconds
Time consumed for prediction: 0.03369 seconds
```

Support Vector Machine Classifier

A support vector machine (SVM) is a supervised machine learning model that uses classification techniques to tackle two-group classification problems. After being fed training data sets with labels for each category, an SVM model may categorize fresh text.

They have two major benefits over more contemporary algorithms like neural networks: they are faster and perform better with less data (in the thousands). As a result, the technique is ideal for text classification problems when access to a database with a few dozen tags on each data set is limited.

The Support Vector model's estimated model accuracy is 39.61%.

```
Scikit-Learn's Support Vector Machine Classifier's prediction accuracy is: 39.61
Time consumed for training: 0.048 seconds
Time consumed for prediction: 0.02836 seconds
```

Strategy 2:

Multilayer Neural Network

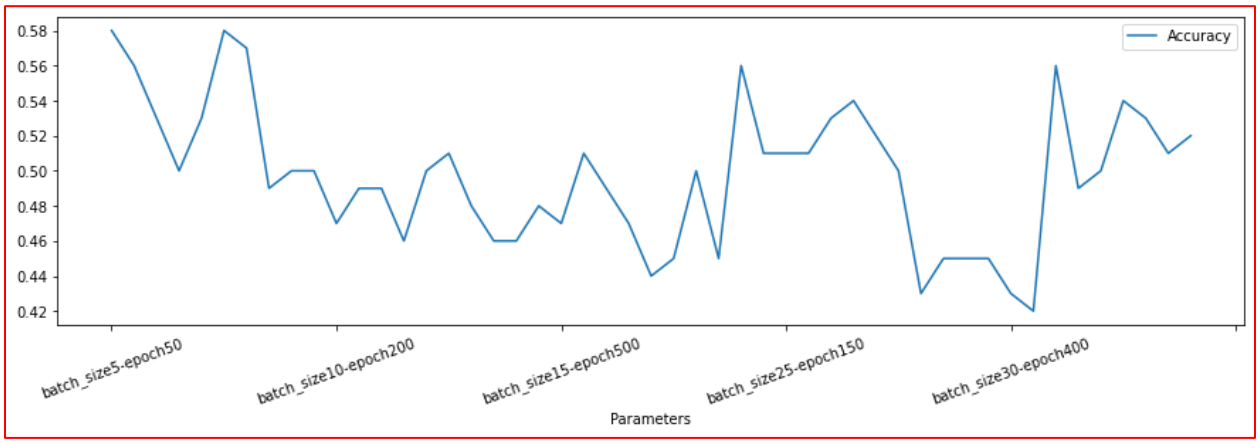
The modeling section started with the exploration of indicators in finer detail. As mentioned in the introduction, we used the metric of the Inter-Quartile Range (IQR) to remove the outliers from the individual indicators. For example, with the outliers present, the mean value for indicator 1 for a good user experience stood around 71.67, while the 75th percentile was still relatively low, at approximately 22.5. We used this approach to section out the outliers for the different indicators based on whether they were a bad user experience or a good one. The next step was consolidating the data between good and bad user experience so that the deployed model could read these differences and develop an algorithm for classifying them.

The data was broken down into an average of the independent indicators separated for good and bad user experiences. The indicators were also normalized using the standard scalar package in python and were loaded into the neural network model. The model selection was made because we wanted the feature selection to be an unbiased process; after that, if more features needed to be added to increase accuracy, that could be done.

The final neural network model had three layers: the input layer, the 'relu' activation function, and the final output layer, which was a 'sigmoid' function. The training data was used to fit the model through batch sizes and epochs.

Evaluation and Deployment

The model was tried and tested using different hyperparameters to obtain the maximum accuracy for training and validation sets. Accuracy took precedence in the selection of the model. Different batch sizes and epochs were used with the default learning rate. The results of the batch size and epoch selection are shown here:



The hyperparameters selected for the model were a batch size of 5 and training epochs of 50.

Results and Accuracy:

Regarding the training accuracy, the model results were 0.65, and the testing and validation accuracy was 0.58.

Classification Model Name	Model Accuracy %
Support Vector Classification	39.61
Random Forest Classification	45.13
K Nearest Neighbors	46.10
XGBoost	48.70
Multilayer Neural Network	58

Future Scope

Finally, a quick overview of all the pertinent facts gathered throughout this thorough investigation is required. Following that, an outlined review is offered to end our assessment and demonstrate recommendations for the future. The data provided is time series. Short Time Fourier Transform (STFT) STFT measures a signal's time-varying frequency content. It has been widely used in time series analysis, such as speech, audio, machine vibration, EKG, and EEG signal processing. Adding STFT values to data might give predictions for models.

LOF (Identifying density-based local outliers) and Isolation Forests combination can be used to detect anomaly (Good / Bad) experiences. Local outlier factor (LOF) values identify an anomaly based on its surrounding. It outperforms the global technique for finding outliers. Because there is no LOF threshold value, identifying a point as an anomaly is entirely up to the user.

Assumptions/Limitations

Massive volumes of available data generated over the previous decade have considerably contributed to deep learning's appeal. This has enabled neural networks to demonstrate their potential since they improve as more data is put into them.

The most well-known drawback of neural networks is their "black box" character. You have no idea how or why NN produced a particular result.

Neural networks often require hundreds, if not millions, of labeled samples compared to classic machine learning techniques. This is a complex problem to solve, as are many machine-learning problems.

Traditional algorithms are also more computationally economical than neural networks. Deep learning algorithms that are cutting-edge.

Appendix (Code)

Git Link: https://github.com/osu-msba/ban5753-fall2022-team-lamda.git

Git Folder: ban5753-fall2022-team-lamda/Final_Code

Title	Code File:	Context
Strategy 1:	Strategy_1_Prep_train-Scaling Strategy_1_Model	Random Forest Classifier XGBoost K Nearest Neighbors Support Vector Machine Classifier
Strategy 2:	Strategy_2_combined_dataset Strategy_2_Model	Multilayer Neural Network

References

[1] Guo Y, Schuurmans D. Semi-supervised Multi-label Classification[C]// Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2012.

[2] Wasnik P P, Phadkule N J, Thakur K D. Fault detection and classification based on semi-supervised machine learning using KNN[C]// 2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET). 2019.

[3] Albashish D, Al-Sayyed R, Abdullah A, et al. Deep CNN Model based on VGG16 for Breast Cancer Classification[C]// 2021 International Conference on Information Technology (ICIT). 2021.

Acknowledgment

We would like to thank Professor Chakraborty Goutam and Dr. Venu Lolla (Guest Faculty), who guided us throughout this project.