BILINGUAL PUBLISHING GROUP

Artificial Intelligence Advances

https://journals.bilpubgroup.com/index.php/aia

ARTICLE

Users' Evaluation of Traffic Congestion in LTE Networks Using Machine Learning Techniques

***Bamidele Moses Kuboye*[1*], *Adedamola Israel Adedipe*[1], *Segun Victor Oloja*[2], *Olanrewaju Ayodeji Obolo*[2]**

[1] *Department of Information Technology, Federal University of Technology, P.M.B. 704, Akure, Nigeria*

[2] *Department of Mathematics and Computer Sciences, University Medical Sciences, P.M.B. 536, Ondo, Nigeria*

ABSTRACT

Over time, higher demand for data speed and quality of service by an increasing number of mobile network subscribers has been the major challenge in the telecommunication industry. This challenge is the result of an increasing population of the human race and the continuous advancement in the mobile communication industry, which has led to network traffic congestion. In an effort to solve this problem, the telecommunication companies released the Fourth Generation Long Term Evolution (4G LTE) network and afterward the Fifth Generation Long Term Evolution (5G LTE) network that laid claims to have addressed the problem. However, machine learning techniques, which are very effective in prediction, have proven to be capable of great importance in the extraction and processing of information from the subscriber's perceptions about the network. The objective of this work is to use machine learning models to predict the existence of traffic congestion in LTE networks as users perceived it. The dataset used for this study was gathered from some students over a period of two months using Google Forms and thereafter, analysed using the Anaconda machine learning platform. This work compares the results obtained from the four machine learning techniques employed which are k-Nearest Neighbour, Support Vector Machine, Decision Tree and Logistic Regression. The performance evaluation of the ML techniques was done using standard metrics to ascertain the real existence of congestion. The result shows that k-Nearest Neighbour outperforms all other techniques in predicting the existence of traffic congestion. This study therefore has shown that the majority of LTE network users experience traffic congestion.

Keywords: Traffic congestion; Fourth generation (4G); Long term evolution (LTE); Machine learning techniques; KNN; SVM; Decision tree; Logistic regression; Subscribers

*CORRESPONDING AUTHOR:

Bamidele Moses Kuboye, Department of Information Technology, Federal University of Technology, P.M.B. 704, Akure, Nigeria; Email: bmkuboye@futa.edu.ng

ARTICLE INFO

Received: 8 February 2023 | Revised: 20 March 2023 | Accepted: 22 March 2023 | Published Online: 31 March 2023

DOI: https://doi.org/10.30564/aia.v5i1.5452

CITATION

Kuboye, B.M., Adedipe, A.I., Oloja, S.V., et al., 2023. Users' Evaluation of Traffic Congestion in LTE Networks Using Machine Learning Techniques. Artificial Intelligence Advances. 5(1): 8-24. DOI: https://doi.org/10.30564/aia.v5i1.5452

1. Introduction

Due to the increasing population of the human race and the continuous advancement in mobile communication technology, the telecommunication industry has witnessed a drastic increase in the number of mobile network subscribers. These mobile subscribers demand higher data rates and quality of service, which has made network service providers, adopt advanced technology, the fourth generation (4G) mobile network, which in turn has given the subscribers some sort of improved experience on the network. Mobile communication technology has undergone a sequence of successive generations before the emergence of the fourth generation (4G), which includes the first generation (1G), the second generation (2G), and the third generation (3G) [1]. Presently, the fifth generation is already in use in most parts of the world but is yet to be launched in Nigeria.

Third Generation Partnership Project (3GPP) developed 4G LTE, a packet-optimized radio access technology, to provide high-speed, low-latency mobile wireless connectivity over long distances, with support for the deployment of bandwidth from 1.25 MHz to 20 MHz and flexible spectrum bands from 450 MHz to 4.5 GHz [2]. According to Kuboye [3], the Nigeria Communication Commission (NCC) has licenced several telecommunications firms to offer 5th-generation broadband services to their subscribers, in order to meet the needs of their subscribers in terms of connection speed, and these firms have all claimed success in the effective deployment of broadband services (4G LTE).

The World Health Organization (WHO) declared a global pandemic in January 2020 due to the 2019 coronavirus disease outbreak (COVID-19). The epidemic has had a huge influence on all industries around the world, posing substantial risks [4].

In line with this declaration by WHO, the president of Nigeria declared absolute lockdown in some Nigerian states, including Lagos State, Ogun State, and the Federal Capital Territory, Abuja, as part of its early response to the COVID-19 outbreak [5,6]. As a result of the lockdown, subscribers began using various digital channels to carry out activities and routines such as teleconferencing, teleseminars, telecommunicating, online teaching and learning, entertainment, and social media interactions. Due to this lockdown, people turned to WhatsApp, Telegram, Zoom, and Google Meets for daily activities, thereby, causing the network to become more congested [7]. The use of these digital channels has increased significantly, owing to this lockdown regulation, which has made the network more and more congested. The outbreak of the pandemic and lockdown caused a spike in data traffic just as subscribers began to adopt digital channels for the majority of their activities and routines including communication, entertainment and social media engagements [8].

Machine Learning is very effective in prediction and can solve these challenges faced by network providers through the extraction and processing of information from the subscribers' perceptions about the network so as to let the network providers know the performance of their network and consequently take proactive measures to solve the problems observed. Machine Learning has progressed as a discipline to the point where it now allows a wireless network to learn and extract knowledge from data by interacting with it [9]. The field of machine learning and communication technology is becoming increasingly intertwined to the extent that when modern machine learning methods are coupled with today's communication systems, they can generate a massive amount of traffic data, which can be used to improve the design and administration of networks and communication components [10].

This work therefore aims to evaluate users' perception of traffic congestion in LTE networks using machine learning techniques. The machine learning techniques used are K-Nearest Neighbour, Support Vector Machine, Decision Tree and Logistic Regression. Thereafter, a comparison of the result of the techniques using standard performance evaluation metrics was done. The rest of this paper is structured as follows: Section 2 reviews related literature; Section 3 described the methodology being adopted while the presentation of results and discussion of the findings is done in Section 4 and Section 5 con-

cludes the study.

2. Review of related literature

Stepanov et al. [11] presented applying machine learning to LTE traffic prediction. They compared Bagging, Random Forest and Support Vector Machine (SVM). The motivation for this study was the urgent requirement to employ cutting-edge techniques for the management of massive information flows brought on by the exponential increase in mobile network traffic caused by the proliferation of users and their associated gadgets. In this study, machine learning techniques including Random Forest, Bagging, and SVM were used to forecast LTE network traffic. This research compared the effectiveness of Bagging, Random Forest, and Support Vector Machines using a dataset titled "predict traffic of LTE network", collected from Kaggle, in which Bagging performed exceptionally well. Alekseeva et al. [12] studied the comparison of machine learning techniques applied to traffic prediction of real wireless networks. Four more machine learning techniques including, Linear Regression, Huber Regression, Bayesian Regression, and Gradient Boosting, were added to the work of Stepanov et al. [11] in this study. Evidence from this research showed that Gradient Boosting provides the highest quality predictions because of its highly effective data determination. For linear models, the Huber loss function was found to optimise the model parameter more effectively, as shown by the performance evaluation.

Khatouni et al. [13] presented a machine-learning approach that tries to predict the latency in a real operational 4G network. Predicting and studying network delay required a massive dataset with over 238 million latency measurements from three distinct commercial mobile service providers. The described solution transformed the latency prediction issue into a multi-label classification problem by flattening the Round Trip Time (RTT) data into many bins. Predictions of RTT class from specified characteristics were made using SVM, Decision Tree, and Logistic Regression. Grid search was used to fine-tune the model's performance by adjusting its hyperparameters, and K-fold cross-validation was used to ensure the model was accurate. The result showed that the decision tree which has an accuracy of 74.3% performed better than SVM and Logistic Regression with an accuracy of 66.4% and 60.9% respectively. Kuboye et al. [7] evaluated the existence of traffic congestion in LTE networks using Convolutional Neural Network (CNN) and Long Short-Term Memories (LSTMs) as Deep Learning techniques. The result showed that LSTM had 82.2% prediction accuracy as against the 76.8% prediction accuracy of CNN. The limitation of this research work was that the dataset used was small and not well suited for any Deep Learning algorithms because they require a very large dataset to make a better prediction and be able to capture the feelings of subscribers accurately.

Fiandrino et al. [14] presented a machine learning-based framework for optimizing the operation of future networks. The capacity of ML tools to manage very complicated systems is what prompted this study; this ability makes ML tools well-suited for managing highly dynamic wireless networks and enables them to make smarter judgement. To characterize traffic characteristics and forecast future traffic demands, the study suggested an ML-based system that instantiated ML pipelines. In comparison to more conventional methods, the results showed a considerable decrease in packet latency. Hassan et al. [15] studied machine learning approach to achieving energy efficiency in relay-assisted LTE-A downlink system. Energy efficiency (EE) is a motivating force in this study because of its recent rise to prominence as an important design parameter for mobile devices. The study attempted to find an EE, or an acceptable compromise between throughput and equity. The study presented many methods for allocating Resource Blocks (RBs) in Long Term Evolution Advanced (LTE-A) networks that make use of relays. A K-means clustering strategy was implemented to remove the dependence of RB and power allocation on relay deployment and user association. To lessen the computational cost of RB allocation, we present a two-stage Neural Network (NN) method that employs unsupervised learning for power allocation.

Also, this study looked at the effects of employing single and multiple L3 relays on EE and throughput.

Li et al. [16] presented the learning and prediction of application-level traffic data in cellular networks. The research set out to find a way to accurately analyse, characterise, and predict mobile traffic using information collected from cellular network providers at the application level. Results showed that some temporally-stable modelled properties and spatial sparsity in traffic statistics exist universally at a service or application level. To better understand these factors, we built a novel traffic prediction framework and designed a dictionary-learning-based alternating direction approach to handling them. The simulation results demonstrated that the proposed framework has the potential to offer a coherent answer for prediction and significantly contribute to addressing issues with modelling and forecasting. Samek et al. [10] studied the convergence of machine learning and communications. The need for innovative machine learning techniques for large data analytics in communication networks prompted this study. This research exploited this understanding of external or internal services by extracting relevant information from network data while taking limited communication resources into consideration. This research further surveyed the application of machine learning to the field of communication at large. The result proves that DNNs are the most efficient way to minimise the expected error over uncertainty, as they don't waste time and resources searching exhaustively across exponential candidate networks.

Zaidi [17] studied nearest neighbour methods and their applications in the design of 5G and beyond wireless networks. A driving force for this study was the prevalence of classification issues in the planning and analysis of today's wireless systems. Using nearest neighbour classifications to address the significant difficulties in communication analysis. This study's contributions can be broken down into two categories: Firstly, an overview of the theoretical and algorithmic framework for solving NN search and classification problems, and secondly, the identification of key emerging scenarios related to 5G and beyond wireless networks that posed a particular classification challenge. The study also included a thorough summary of the problem's background and how various research have framed it within the NN framework. Khan et al. [18] reviewed the survey and taxonomy of clustering algorithms in 5G. The phenomenal expansion of a variety of UEs was the motivation for this study due to the enormous amounts of data they create. The work's stated goal was to enhance network performance by using clustering to better organise network topology and summarise data. A clustering framework is provided alongside a taxonomy for clustering qualities that include aims, difficulties, metrics, characteristics, and performance indicators.

3. Model architecture

The system architectural diagram for our proposed traffic congestion prediction model for this study is presented in **Figure 1**. This model was divided into several phases namely; the data preparation, the data splitting, the modelling, the classification, the model evaluation and tuning, as well as, the result.

In the model evaluation and tuning phase, the performances of all the machine learning algorithms were evaluated based on metrics like accuracy, F1 score, and others, whereby poor-performing algorithms were further tuned in a process called hyper-parameter optimization. In the result phase, the performances of the algorithms were compared, and the best-performing algorithm was used to make the final prediction.

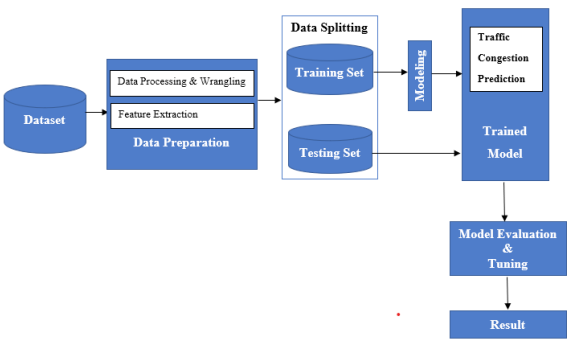

Figure 1. System architecture for the traffic congestion prediction model.

3.1 Data collection

Due to the sensitivity of the data, collection of raw data from network providers was not possible as they are not willing to release them. Hence, the data were gathered from an online survey using Google Forms. The survey was carried out from the 30th of April, 2022 up until the 20th of June, 2022. The survey consisted of 1 short text question, 1 checkbox question, and 10 multiple-choice questions. The aim was to survey university students in Nigeria, with a population size of over 2.1 million according to the statistics gotten from the Statista research department and the Nigerian Tribune [19,20]. The sampling method employed in gathering the data from the university students in Nigeria was a non-probability sampling method, which involves non-random selection based on the combination of convenience sampling and snowball sampling. Convenience sampling is a sampling that includes the students who are most accessible, while snowball sampling is used to get in contact with students via other students [21]. The survey was conducted online, and it took the students no more than 2 minutes to complete it anonymously. The sample size was supposed to be 275 students, going by the statistics from Statista and the Nigerian Tribune. Fortunately, 310 students responded to the survey.

3.2 Data preparation

In the data preparation phase, the data gathered from an online survey were cleaned, wrangled, curated, and prepared before any machine learning techniques are applied.

Data preprocessing

The data gathered from the online survey are usually presented in its raw or slightly processed form, which may not be appropriate for the proposed traffic congestion prediction model. Therefore, the data would have to undergo some pre-processing tasks. Examining the raw form of the survey data, it was observed that all 12 questions for which responses were generated represented the attributes of the dataset. The responses were downloaded into an Excel spreadsheet where they can be easily modified and transformed. Each of these questions was transformed into various attribute names. The dataset was then exported into a comma-separated values (CSV) file format, which could be easily read and handled by the libraries and packages in Jupyter Notebook. The dataset contains 310 instances with 11 conditional features and 1 target feature, which are all categorical. The target feature is "Traffic_Congestion", which is a binary decision on whether traffic congestion is experienced or not. The target class distribution was 189 for "Yes" and 121 for "No" as shown in **Figure 2**. The description of the attributes in the traffic congestion dataset is shown in **Table 1**. The dataset has no missing values because the survey was filled out completely. Since the attributes are all categorical, the attributes were encoded into dummy values using the LabelBinarizer and OneHotEncoder functions from scikit-learn as shown in **Figure 3**.

```
In [6]: # Checking the distribution of target feature
        traffic_df.Traffic_Congestion.value_counts()

Out[6]: Yes    189
        No     121
        Name: Traffic_Congestion, dtype: int64
```

Figure 2. Distribution of the target feature.

Feature extraction

For the model to generalise extremely well on the data and minimise overfitting, it was necessary to choose some features while dropping others based on feature relevance and quality. Better model performance, reduced computational and model training time, and a deeper grasp of the significance of different features in the data are further compelling arguments in favour of feature extraction. To have a clean and interpretable dataset, some attributes from the original dataset were dropped after checking for missing values in **Figure 4**, as shown in **Figure 5** and **Table 2**. **Table 3** shows the selected and used attributes for the traffic congestion prediction model.

Table 1. Description of the attributes in the traffic congestion dataset.

S/N	Attribute name	Attribute type	Description
1	University	Categorical	Abbreviation of the institutions attended by the students
2	Level	Categorical	The level of the students
3	Network_Providers	Categorical	The names of the network service providers
4	4G_Simcard	Categorical	The type of sim card being used
5	Data_Usage	Categorical	Data usage affirmation
6	Data_Usage_Period	Categorical	Period/time of data usage
7	Internet_Surfing_period	Categorical	Period/time of surfing the internet
8	Internet_Surfing_Rate	Categorical	The rate at which the internet was being surfed
9	Download_Period	Categorical	The period at which downloading was done
10	Download_Speed	Categorical	The speed of the network when downloading
11	Data_Consumption_Platform	Categorical	The various platforms that consume most data
12	Traffic_Congestion	Categorical	Experience of traffic congestion on the network

```
In [5]: # Info
        traffic_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310 entries, 0 to 309
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   University                 310 non-null    object
 1   Level                      310 non-null    object
 2   Network_Providers          310 non-null    object
 3   4G_Simcard                 310 non-null    object
 4   Data_Usage                 310 non-null    object
 5   Data_Usage_Period          310 non-null    object
 6   Internet_Surfing_Period    310 non-null    object
 7   Internet_Surfing_Rate      310 non-null    object
 8   Download_Period            310 non-null    object
 9   Download_Speed             310 non-null    object
 10  Data_Consumption_Platform  310 non-null    object
 11  Traffic_Congestion         310 non-null    object
dtypes: object(12)
memory usage: 29.2+ KB
```

Figure 3. Information about the dataset.

```
In [11]: # Checking for missing values
         traffic_df.isnull().sum()

Out[11]: University                 0
         Level                      0
         Network_Providers          0
         4G_Simcard                 0
         Data_Usage                 0
         Data_Usage_Period          0
         Internet_Surfing_Period    0
         Internet_Surfing_Rate      0
         Download_Period            0
         Download_Speed             0
         Data_Consumption_Platform  0
         Traffic_Congestion         0
         dtype: int64
```

Figure 4. Checking missing values in the dataset.

```
In [13]: # Dropping irrelevant columns
         traffic_df.drop(columns=["University", "Level", "Data_Usage", "Data_Consumption_Platform"], inplace=True)
```

Figure 5. Dropped features in the dataset.

3.3 Data splitting

Well-prepared data coming from the data preparation phase was divided into a training set and a testing set in the data splitting phase. Data splitting is commonly used in machine learning to split data into a training, testing, or validation set. This approach allows us to find the model hyper-parameter and also estimate the generalisation performance. In this research, the dataset was divided into training and testing sets at a proportion of 80% and 20% respectively. The training dataset was used for classification, while the test dataset was used to evaluate classification correctness. 'X' was used to represent the training features and 'y' to represent the target feature as shown in **Figure 6**.

Training set

The training set is 80% of the complete traffic congestion dataset that was used to build the ML model. The models observe and learn from this data and optimise its parameters, such that 248 rows of the dataset were used as a training set to develop four models that were supposed to make congestion predictions as shown in **Figure 6**.

Testing set

The testing set is a sample of data used to objectively assess how well a final model fits the training dataset. It is only applied once the model has finished training with the training set. Thus, the testing set is the one that is used to simulate the kind of circumstances that will be experienced after the model is made available for use in real time. In this research, 20% of the traffic congestion dataset was used as a testing set, such that 62 rows of the data were used to test the performance of the model as shown in **Figure 6**.

Table 2. Dropped attributes in the traffic congestion dataset.

S/N	Attribute name	Reason for removal
1	University	If further processed and used, it will shift the research away from its focus
2	Level	If further processed and used, it will shift the research away from its focus
3	Data_Usage	All the respondents (students) use data
4	Data_Consumption_Platform	If further processed and used, it will shift the research away from its focus

Table 3. Selected and used attributes in the traffic congestion dataset.

S/N	Attribute name	Attribute type	Description
1	Network_Providers	Categorical	The names of the network service providers
2	4G_Simcard	Categorical	The type of sim card being used
3	Data_Usage_Period	Categorical	Period/time of data usage
4	Internet_Surfing_period	Categorical	Period/time of surfing the internet
5	Internet_Surfing_Rate	Categorical	The rate at which the internet was being surfed
6	Download_Period	Categorical	The period at which downloading was done
7	Download_Speed	Categorical	The speed of the network when downloading
8	Traffic_Congestion	Categorical	Experience of traffic congestion on the network

```
In [26]: # Splitting of columns into X and y
         X = traffic_df.drop("Traffic_Congestion", axis=1)
         y = traffic_df["Traffic_Congestion"]

In [27]: # Train test split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=10)

In [28]: # Shapes of the train and test variable
         print(X_train.shape)
         print(X_test.shape)
         print(y_train.shape)
         print(y_test.shape)

         (248, 22)
         (62, 22)
         (248,)
         (62,)
```

Figure 6. Splitting the training and testing features.

3.4 Modelling phase

In the modelling phase, machine learning algorithms including Logistic Regression, K-Nearest Neighbours, Support Vector Machine and Decision Trees employed in the training set. The trained model was used to predict the unseen data, which is the testing set, in the classification phase.

Logistic regression

Logistic regression involves assigning numerical values to each attribute and then summing those values together. Then the sigmoid function, which returns binary values, is applied to the result. Logistic regression is used to produce the coefficients for predicting the log and its transformed probability [22]. The probability $p(y = 1|x_i)$ of logistic regression is estimated in two steps. The first step estimates a linear regression function $g(x_i)$ based on the input variables from the vector x_i:

$$g(x_i) = \beta_0 + \beta_i . x_i + ... + \beta_j . x_{ij} \quad (1)$$

where β_0 is called the intercept and β_1, β_2, ..., β_j are called the Regression Coefficients of x_1, x_2, ..., x_{ij}, respectively. Regression coefficients characterize the relative importance of each risk factor. A positive regression coefficient indicates an increase in the likelihood of the result due to the risk factor, whereas a negative regression coefficient indicates a reduction in the likelihood of the outcome due to the risk factor. To describe the degree to which a risk factor contributes to the likelihood of an event, the regression coefficient can be expressed as a positive or negative number.

In the second step, the results of the regression function $g(x_i)$ must be transformed to have bounds between 0 and 1, thus, representing the probability $p(y = 1|x_i)$. The transformation of regression results is done by a logistic function. The logistic function f(t) is mathematically expressed as [23]:

$$f(t) = \frac{e^t}{1 + e^t} \quad (2)$$

where e^t is called the time exponential.

Combining regression Equation (1) with logistic Equation (2), we get the formula for logistic regression probability $p(y = 1|x_i)$ prediction:

$$p(y = 1|x_i) = \frac{e^{g(x_i)}}{1 + e^{g(x_i)}} \quad (3)$$

Equation (3) converges to 1 for high positive values of the regression function $g(x_i)$. On the other hand, probability $p(y = 1|x_i)$ goes to 0 for negative values of the regression function $g(x_i)$ [24]. The logistic function is useful because it can take an input of any value from negative infinity to positive infinity, whereas the output is confined to values between 0 and 1.

k-Nearest Neighbour (KNN)

k-Nearest Neighbour is a classification and regression algorithm that does not make any assumptions about the data that need to be distributed [25]. It is frequently used to determine where the task should be classified in the database's unseen instances [26]. It employs a non-parametric technique, which means that instead of extracting a finite number of parameters from the training set, it makes predictions using the entire set [27].

Support vector machine (SVM)

Support Vector Machines (SVMs) as a kind of supervised machine learning are widely employed for classification tasks. To classify data in a high-dimensional feature space, which was created from the original input space using non-linear methods, SVM employs a straightforward linear approach [28]. In other words, input data are transformed into a high-dimensional feature space where the data can be separated linearly. A kernel function k is responsible for transforming the input data into the high-dimensional feature space.

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \quad (4)$$

where $\phi : X \rightarrow H$ is a projection from feature space into high-dimensional feature space.

A hyperplane is afterward used to separate the data in the high-dimensional feature space. The best hyperplane has the greatest separation margins between the two classes. By solving a quadratic optimization problem with constraints, the maximum separation is achieved.

The classification decision function with hyper-plane as a parameter can be mathematically expressed as follows:

$$f(x) = sign\big(\langle w, \phi(x)\rangle + b\big) \quad (5)$$

where w is a solution of quadratic optimization and b is a constant.

Decision tree (DT)

A decision tree (DT) is a non-parametric technique used for both classification and regression. It utilizes the divide-and-conquer approach [26]. Its structure is based on a series of tests on the input attributes, also called internal nodes. It leads to a finite discrete number of replies, which results in more internal nodes, and so on until the final branches are reached [27]. Many measures can be used to determine the best way to split the records, thus, defined in terms of the class distribution of the records before and after splitting. The measures developed for selecting the best split are often based on the degree of impurity of the child nodes [29]. The impunity measures include:

$$Entropy(t) = -\sum_{i=0}^{c-1} \mathrm{p(i|t)log2p(i|t)} \quad (6)$$

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [\mathrm{p(i|t)}]^2 \quad (7)$$

$$Classification\ error(t) = 1 - max\ i\ [p(i|t)] \quad (8)$$

where c is the number of classes and 0 $\log_2$ 0 = 0 in entropy calculations [29].

3.5 Model performance metrics

After classification, there is a need to validate the model built on the testing set of the dataset. There are various measures for evaluating the classifier performance such as Accuracy Score, Classification Report, Confusion Matrix, Precision, F1 Score, Recall, ROC-AUC Curve, and others. Each measure has its merits and demerits. A combination of these is preferable to use rather than a single measure to evaluate the performance of traffic congestion prediction models.

Confusion matrix

The performance of classification models on the test dataset may be described using a square matrix table called the confusion matrix. The confusion matrix provides insight into the areas in which categorization methods succeed and fail. Therefore, count values are used to summarise and categorise the number of accurate and inaccurate forecasts. True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) parameters were obtained through a confusion matrix. The counts of correct and incorrect classifications are then filled into the table as shown in **Table 4**, where all correct classifications lie along the principal diagonal (TP and TN) and the incorrect classifications correspond to numbers of the diagonal (FP and FN).

Table 4. Confusion matrix for two classes.

		Predicted Class	
		Negative (0)	Positive (1)
Actual Class	Negative (0)	TN	FP
	Positive (1)	FN	TP

TP is an outcome where the model predicts the positive class accurately. FP is the situation that the model predicts the positive class inaccurately. TN is the outcome in which the model accurately predicts the negative class. FN is the result that shows the model predicts the negative class inaccurately [30].

Accuracy

Accuracy in classification problems indicates the proportion of the correctly classified cases both positive and negative on a particular dataset. It is the success ratio of TP and TN on specific datasets. It is the measure of how correctly the algorithm classified the unseen instances. An ideal classifier should have a higher degree of accuracy, thus, be expressed mathematically in Equation (9) [30].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (9)$$

Precision

Precision is defined as the ratio of the actual number of positive results to the predicted number of positive results from a classifier. It measured the proportion of all data predicted positive if they were actually positive. An ideal classifier should have a higher degree of precision. It is mathematically expressed in Equation (10) [30,31].

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

Recall

Recall or Sensitivity is used to calculate the pro-

portion of actual positives that are correctly classified. An ideal classifier should have a higher degree of recall. It is mathematically expressed in Equation (11) [30,31].

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

F1 score

F1 score is needed when you want to seek a balance between Precision and Recall, thus, giving its mathematical notation in Equation (12) [30].

$$F1\ score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{12}$$

ROC-AUC

Receiver Operating Characteristics (ROC) and Area Under the Curve (AUC) is a probability curves for displaying the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). The ROC-AUC curve shows the model's capacity to distinguish across classes. In an ideal classifier, FPR must equal zero and TPR must equal one [31].

Cohen's Kappa score

Cohen's Kappa is used to measure the performance of a classification model, thus, given in Equation (13).

$$K = \frac{p_0 - p_e}{1 - p_e} \tag{13}$$

where $\mathbf{p_0}$ is the empiricsal probability of agreement on the label assigned to any sample and $\mathbf{p_e}$ is the expected agreement when both annotators assign labels randomly. The closest the value of the Kappa score to 1, the better for the classifier [30].

4. Results

In this section, a comparison of the four machine-learning algorithms was done based on the output of the Jupyter Notebook for each classifier.

4.1 K-Nearest Neighbours (KNN)

There is a need to search for an optimal value of k before passing the already split dataset into the KNN algorithm, as shown in **Figure 7**. The optimal value of k is a value in which the KNN algorithm performs optimally and predicts accurately. Thereafter, the testing dataset was used to evaluate the performance of the KNN classifier, as shown in **Figure 8** and **Figure 9** showing the confusion matrix and the ROC-AUC curve for the KNN classifier respectively.

```
In [35]: # search for an optimal value of k for KNN
         k_range = range(1, 40)
         error_rate = []
         for k in k_range:
             knn1 = KNeighborsClassifier(n_neighbors=k)
             knn1.fit(X_train, y_train)
             y_pred1 = knn1.predict(X_test)
             error_rate.append(np.mean(y_pred1 != y_test))
```

Figure 7. Searching for the optimal value of k.

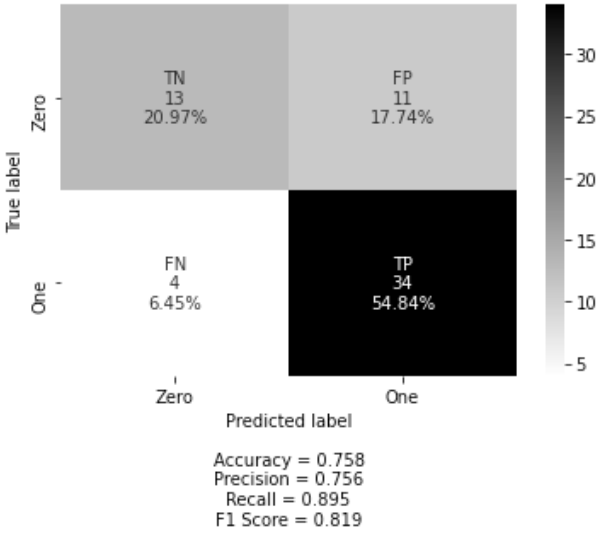

Figure 8. Confusion matrix for the KNN classifier.

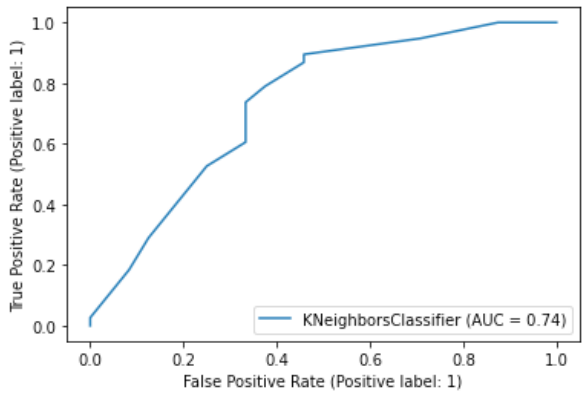

Figure 9. ROC-AUC curve for the KNN classifier.

4.2 Support vector machine (SVM)

There is a need to search for the best parameters in the parameter grid before passing the already

splitted dataset into the SVM algorithm, as shown in **Figure 10**. The best parameters are parameters in which the SVM algorithm performs optimally and predict accurately.

The performance evaluation on a training and testing dataset for the SVM classifier was given in **Figure 11** and **Figure 12** showing the confusion matrix and the ROC-AUC curve for the SVM classifier respectively.

4.3 Logistic regression classifier (LRC)

There is a need to search for the best parameters in the parameter grid before passing the already split dataset into the LRC algorithm, as shown in **Figure 13**. The best parameters are parameters in which the LRC algorithm performs optimally and predict accurately.

The performance evaluation on a training and testing dataset for LRC is shown in **Figure 14** and **Figure 15** showing the confusion matrix and the ROC-AUC curve for the LR classifier respectively.

```
In [56]: # Parameter Grid{dictionary} for C and gamma
         param_grid = {'C':[0.1,1,10,100,1000],'gamma':[1,0.1,0.01,0.001,0.0001]}

In [57]: # Create and Train the GridSearchCV model for SVC
         grid = GridSearchCV(SVC(probability=True), param_grid, scoring='accuracy', cv=10, verbose=2)
         grid.fit(X_train, y_train)
```

Figure 10. GridSearch for SVM classifier.

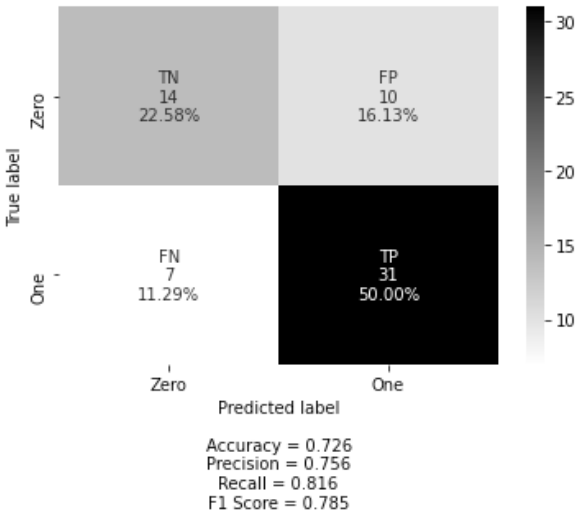

Figure 11. Confusion matrix for the SVM classifier.

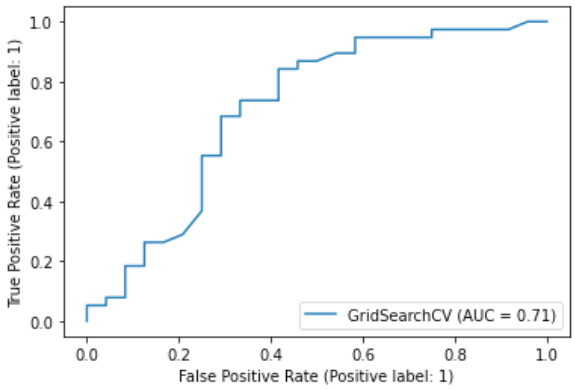

Figure 12. ROC-AUC curve for the SVM classifier.

```
In [70]: # Parameter Grid{dictionary} for C, penalty, and solver
         param_grid1 = {'penalty':['l1', 'l2'], 'C':np.logspace(-3,3,7), 'solver':['newton-cg', 'lbfgs', 'liblinear']}

In [71]: # Create and Train the GridSearchCV model for LRC
         grid1 = GridSearchCV(lrc, param_grid1, scoring='accuracy', cv=10, verbose=2)
         grid1.fit(X_train, y_train)
```

Figure 13. GridSearch for LRC.

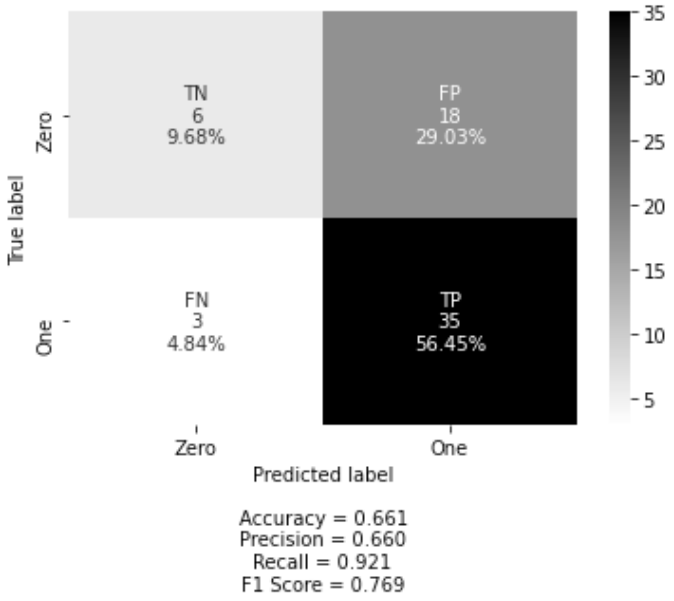

Figure 14. Confusion matrix for the LR classifier.

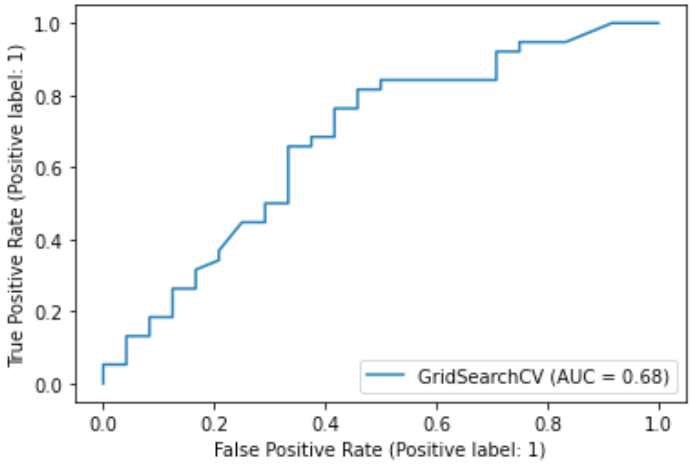

Figure 15. ROC-AUC curve for the LR classifier.

4.4 Decision tree classifier (DTC)

There is a need to search for the best parameters in the parameter grid before passing the already split dataset into the DTC algorithm, as shown in **Figure 16**. The best parameters are parameters in which the DTC algorithm performs optimally and predict accurately.

The performance evaluation on a training and testing dataset for DTC is shown in **Figure 17** and **Figure 18** showing the confusion matrix and the ROC-AUC curve of the DT classifier respectively.

4.5 Model comparison

To evaluate the performance of the best-performing model, the results of all the models employed were compared. **Table 5** shows all the results of each classifier's performances based on Accuracy, Precision, Recall, F1 score, Kappa and AUC.

The performances of all four classifiers were further expressed visually using multiple bar charts as shown in **Figure 19**. The ROC-AUC curves for all four classifiers were plotted together in **Figure 20**.

```
In [84]: # Parameter Grid{dictionary} for max_depth, min_samples_leaf, and criterion
         param_grid2 = {'max_depth':[2, 3, 5, 10, 20], 'min_samples_leaf':[5, 10, 20, 50, 100], 'criterion':['gini', 'entropy']}

In [85]: # Create and Train the GridSearchCV model for DTC
         grid2 = GridSearchCV(DecisionTreeClassifier(random_state=10), param_grid2, scoring='accuracy', cv=10, n_jobs=-1, verbose=2)
         grid2.fit(X_train, y_train)
```

Figure 16. GridSearch for DTC.

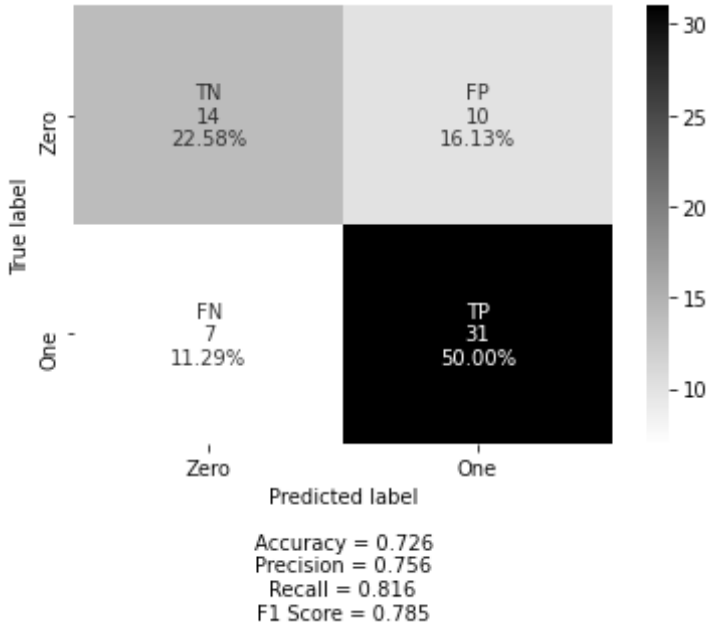

Figure 17. Confusion matrix for the DT classifier.

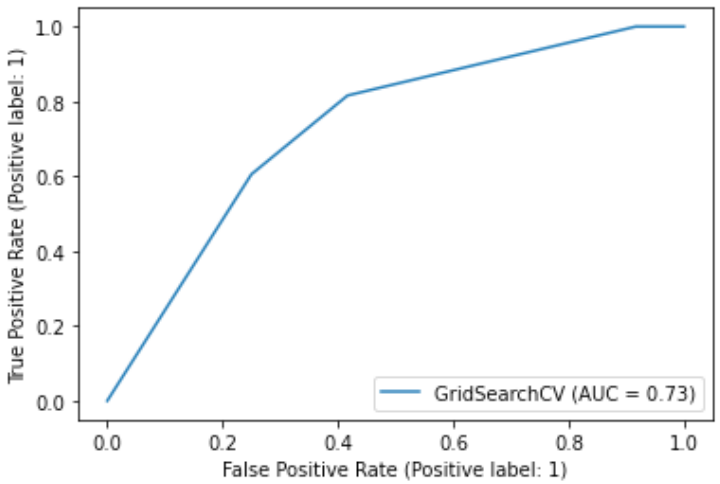

Figure 18. ROC-AUC curve for the DT classifier.

Table 5. Results for the four classifiers.

Models	Accuracy	Precision	Recall	F1 score	Kappa	AUC
K Nearest Neighbours	0.76	0.76	0.89	0.82	0.46	0.74
Support Vector Machine	0.73	0.76	0.82	0.78	0.41	0.71
Logistic Regression	0.66	0.66	0.92	0.77	0.19	0.68
Decision Tree	0.73	0.76	0.82	0.78	0.41	0.73

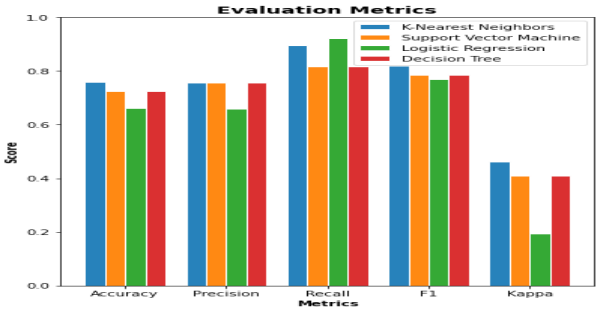

Figure 19. Performances of the four classifiers on the traffic congestion dataset.

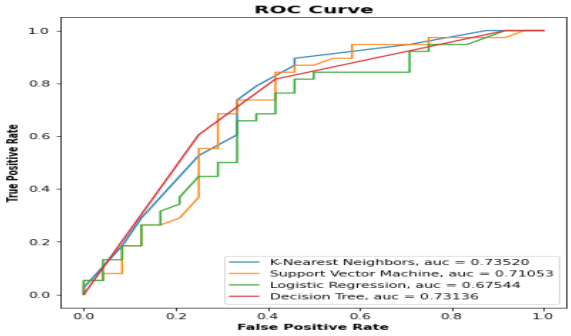

Figure 20. ROC-AUC curves for the four classifiers.

4.6 Discussion of the result

Table 5 shows that KNN has the highest overall accuracy, the highest F1 score, the highest Kappa score and the highest AUC amongst all the classifiers. In contrast, Logistic Regression is the lowest amongst all the classifiers for Accuracy, Precision, F1 score, Kappa score and AUC, whereas it is the highest in Recall with 92% (0.921). It means that Logistic Regression is the most sensitive classifier in this classification problem. KNN, SVM, and Decision Tree all have the same Precision of 76% (0.76). SVM and Decision Tree have the same Accuracy, Precision, Recall, F1 score, and Kappa but have a slightly different AUC, with SVM AUC of 0.710 and Decision Tree AUC of 0.731. **Figure 19** shows a graphical representation of the performances of the four models used in this research on the Traffic Congestion dataset. **Figure 20** combines all the ROC-AUC curves of the four classifiers into a single representation. It is clear from the figure that KNN has the highest AUC of 0.7352, closely followed by Decision Tree AUC of 0.73136, followed by SVM AUC of 0.71053, and lastly, Logistic Regression AUC of 0.67544. For comparison purposes, according to all of these performance evaluation metrics, one would agree that KNN is the best-performing classifier. So, their performance is in the order of KNN, Decision Tree, SVM, and Logistic Regression. This research corroborates with research conducted by Kuboye et al. [7] that despite the promising better Internet services that necessitated the rollout of LTE, con-

gestion is still being experienced since all the classifiers give a higher value. The reason is that more subscribers are now being accommodated. Even though the focus of the research conducted by Khatouni et al. [13] differed slightly from the focus of this research, it is further supported in this research that users of LTE services experienced traffic congestion. Furthermore, by contrasting the research of Stepanov et al. [11] and Alekseeva et al. [12] with this research and critically observing the performance of the various machine learning algorithms used to predict traffic congestion, it can be further substantiated in this research that traffic congestion is being observed in LTE networks. The reason for this is that the traffic generated is rapidly increasing as more subscribers embrace the technology on a daily basis.

5. Conclusions

KNN, SVM, Decision Tree, and Logistic Regression machine learning techniques were considered in this study for traffic congestion prediction. The four classifiers were compared using the users' perception data gathered from an online survey, which was carried out using Google Forms. The result of this study showed that KNN proved to be more accurate in predicting the existence of traffic congestion as compared to other classifiers being employed. Predictions carried out with these techniques showed that the majority of LTE network users experience traffic congestion. It is highly recommended that telecommunication companies improve their network bandwidth so as to minimise traffic congestion. Future studies would consider online machine learning techniques that can continuously read data coming from network operators so as to obtain the relevant features and perform the traffic congestion prediction in real-time to assist the traffic providers to employ mechanisms that can reduce traffic congestion to the barest minimum.

Conflict of Interest

There is no conflict of interest.

References

[1] Kuboye, B.M., 2019. Long term evolution (LTE) network evaluation in the south-west region of Nigeria. European Journal of Engineering and Technology Research. 4(3), 86-92.

[2] Tchao, E.T., Gadze, J.D., Agyapong, J.O., 2018. Performance evaluation of a deployed 4G LTE network. (IJACSA) International Journal of Advanced Computer Science and Applications. 9(3).

[3] Kuboye, B.M., 2017. Evaluation of broadband network performance in Nigeria. International Journal of Communications, Network and System Sciences. 10(9), 199-207.

[4] Kim, J.H., 2021. Data-driven approach using machine learning for real-time flight path optimization [PhD thesis]. Atlanta: Georgia Institute of Technology.

[5] Amzat, J., Aminu, K., Kolo, V.I., et al., 2020. Coronavirus outbreak in Nigeria: Burden and socio-medical response during the first 100 days. International Journal of Infectious Diseases. 98, 218-224.

[6] Dan-Nwafor, C., Ochu, C.L., Elimian, K., et al., 2020. Nigeria's public health response to the COVID-19 pandemic: January to May 2020. Journal of Global Health. 10(2), 1-9.

[7] Kuboye, B.M., Aratunde, T.O., Gbadamosi, A.A., 2021. Users' evaluation of traffic congestion in LTE networks using deep learning techniques. International Journal of Computer Applications. 975, 8887.

[8] Idris, A., 2020. MTN Nigeria Records a Spike in Data Traffic, But Voice Revenue is Still King [Internet]. TechCabal [cited 2022 Sep 13]. Available from: https://techcabal.com/2020/04/30/mtn-q1-2020-financial-report-voice-data-revenue/

[9] Morocho-Cayamcela, M.E., Lee, H., Lim, W., 2019. Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions. IEEE Access. 7, 137184-137206.

[10] Samek, W., Stanczak, S., Wiegand, T., 2017. The convergence of machine learning and communications. arXiv:1708.08299.
DOI: https://doi.org/10.48550/arXiv.1708.08299

[11] Stepanov, N., Alekseeva, D., Ometov, A. (editors), et al., 2020. Applying machine learning to LTE traffic prediction: Comparison of bagging, random forest, and SVM. 2020 12th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT); 2020 Oct 05-07; Brno, Czech Republic. USA: IEEE. p. 119-123.

[12] Alekseeva, D., Stepanov, N., Veprev, A., et al., 2021. Comparison of machine learning techniques applied to traffic prediction of real wireless network. IEEE Access. 9, 159495-159514.

[13] Khatouni, A.S., Soro, F., Giordano, D. (editors), 2019. A machine learning application for latency prediction in operational 4g networks. 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM); 2019 Apr 8-12; Arlington, VA, USA. USA: IEEE. p. 71-74.

[14] Fiandrino, C., Zhang, C., Patras, P., et al., 2020. A machine-learning-based framework for optimizing the operation of future networks. IEEE Communications Magazine. 58(6), 20-25.

[15] Hassan, H., Ahmed, I., Ahmad, R., et al., 2019. A machine learning approach to achieving energy efficiency in relay-assisted LTE-A downlink system. Sensors. 19(16), 3461.

[16] Li, R., Zhao, Z., Zheng, J., et al., 2017. The learning and prediction of application-level traffic data in cellular networks. IEEE Transactions on Wireless Communications. 16(6), 3899-3912.

[17] Zaidi, S.A.R., 2021. Nearest neighbour methods and their applications in the design of 5G and beyond wireless networks. ICT Express. 7(4), 414-420.

[18] Khan, M.F., Yau, K.L.A., Noor, R.M., et al., 2020. Survey and taxonomy of clustering algorithms in 5G. Journal of Network and Computer Applications. 154, 102539.

[19] Statista Research Department, 2022. Number of Undergraduate Students at Universities in Nigeria as of 2019, by Gender and Discipline [Internet]. Statista [cited 2022 Jun 20]. Available from: https://www.statista.com/statistics/1262928/number-of-undergraduate-students-at-universities-in-nigeria-by-gender-and-discipline/

[20] Idoko, C., 2021. 2.1 Million Students Studying in Nigerian Universities [Internet]. Nigerian Tribune [cited 2022 Jun 20]. Available from: https://tribuneonlineng.com/2-1-million-students-studying-in-nigerian-universities%E2%80%95-nuc/

[21] Yadav, S.K., Singh, S., Gupta, R., 2019. Biomedical statistics, a beginner's guide. Springer: Singapore. pp. 71-83.

[22] Patel, A., 2018. Machine Learning Algorithm Overview [Internet] [cited 2020 May 9]. Available from: https://medium.com/ml-research-lab/machine-learning-algorithm-overview-5816a2e6303

[23] Kuhn, M., Johnson, K., 2013. Applied predictive modeling. Springer: New York. pp. 13.

[24] Polena, M., 2017. Performance analysis of credit scoring models on lending club data [Master's thesis]. Prague: Charles University.

[25] Sun, Y., Peng, M., Zhou, Y., et al., 2019. Application of machine learning in wireless networks: Key techniques and open issues. IEEE Communications Surveys and Tutorials. 21(4), 3072-3108.

[26] Mohamed, A.E., 2017. Comparative study of four supervised machine learning techniques for classification. International Journal of Applied. 7(2).

[27] Guerra, T., 2018. Machine Learning Based Han-

dover Management for LTE Networks with Coverage Holes [Internet]. Available from: https://repositorio.ufrn.br/bitstream/123456789/26678/1/Machinelearningbased_Guerra_2018.pdf

[28] Karatzoglou, A., Meyer, D., Hornik, K., 2006. Support vector machines in R. Journal of Statistical Software. 15, 1-28.

[29] Awad, M., Khanna, R., 2015. Efficient learning machines: Theories, concepts, and applications for engineers and system designers. Springer Nature: Berlin. pp. 268.

[30] Grandini, M., Bagli, E., Visani, G., 2020. Metrics for multi-class classification: An overview. arXiv:2008.05756.
DOI: https://doi.org/10.48550/arXiv.2008.05756

[31] Marabad, S., 2021. Credit card fraud detection using machine learning. Asian Journal for Convergence in Technology. 7(2), 121-127.