measurement tool called WebLAR (Web latency and Rendering) that can measure web latency and QoE in the cellular networks. They found out that TCP connect time and Time To First Byte (TTFB) in LTE network are 160% and 30% longer than fixed line network. Their result showed that DNS look up time varies significantly with the TCP connection time of the websites they studied across Mobile Network Operators but the difference between page load time and above the fold time across operators was not significant. But this study was carried out on only eight websites. For facebook and other social media application.

Boz *et al*, (2019) in another work titled "Youtube and facebook QoE in mobile Broadband networks", studied the performance of the popular youtube and facebook applications in mobile broadband networks from the end user's perspective and presented the first result on the evaluation of different 3.5G network conditions on youtube and facebook from the end users perspective, considering everyday life web usage scenario.

Lycett & Radwan, (2019) identified significant challenges of developing models that uses the quality of experience of web applications and in trying to solve the problem, presented a novel model that integrates factors through key performance indicators and key quality indicators. They mapped the metrics and incorporated them into a correlation model that assesses the Quality of Experience of web applications. The resultant data from mappings was used as input of the proposed model to develop artefacts that quantify and predict user's experience.

Casas *et al*, (2017) also studied QoE of popular applications in smartphones. They addressed the problem of QoE monitoring, assessment and prediction in cellular networks relying on in-smartphone QoS traffic measurement and QoE crowdsourced feedback. They developed system for predicting QoE in smartphones for popular applications in a distributed manner using only in-smartphone passive traffic measurement.

Alreshoodi and Woods, (2013) presented a brief review of some existing correlation models which attempt to map Quality of Service (QoS) to Quality of Experience (QoE) for multimedia services. This contribution analysed a number of previous attempts and optimisation techniques that can reliably compute the weighting coefficients for the QoS/QoE mapping.

Vasilev *et al*, (2018) used machine learning techniques to demonstrate how QoS metrics can be utilized to accurately estimate and predict key QoE factors. Their focus was mostly on the stall label QoE factors as it is the hardest to predict. But to improve QoE prediction, new features specific to video profiling was designed and can be measured by QoS monitoring systems.

Wuruola, (2018) in using machine learning to predict quality of experience, developed a model called Quality of Experience of Web Applications (QoEWA) by noting that the relationship between objective and subjective factors was a challenge and addressed this by introducing machine learning as a means for QoEWA model to predict and evaluate subjective data dynamically. But user feedback on this system is generally limited and MOS is generally time consuming and expensive to process.

3.0 Materials and Methods

The architecture of the quality of experience model is shown in figure 1 and it comprised of four important modules: The user's equipment, feature extraction, quality of experience server, model training and model evaluation.

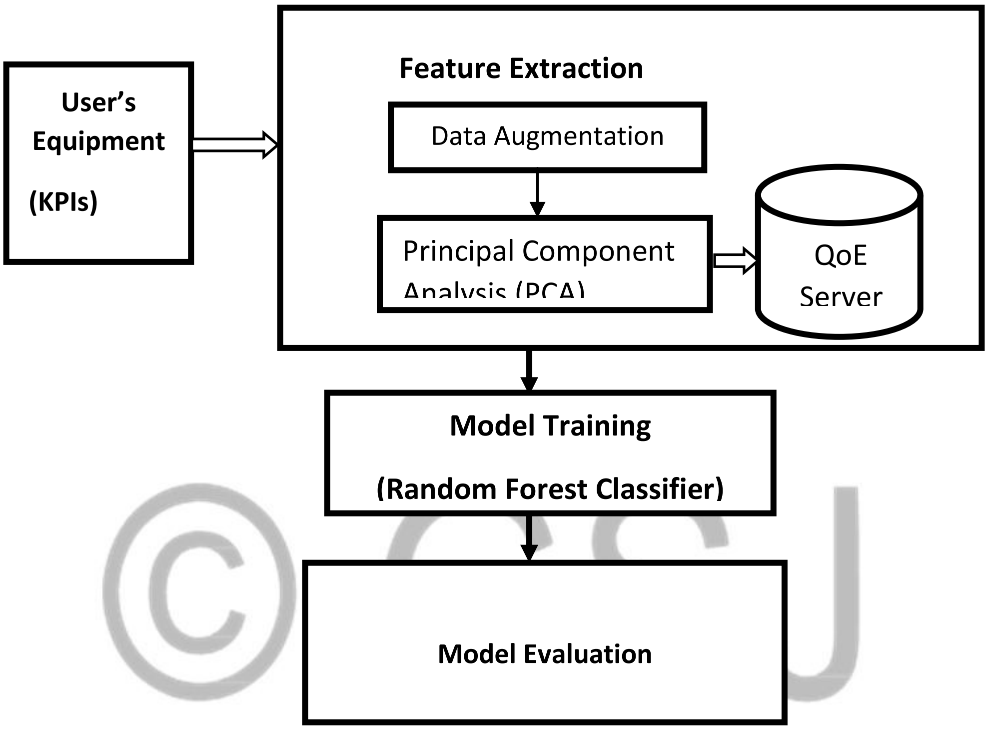

Figure 1: Architecture of the Quality of Experience Predictive Model

The Data capture / Users Equipment is responsible for data collection from the user's device as soon as the network is launched. The user interacts with the system by first locating the directory where the data is stored. This Quality of Service (QoS) dataset which contains a total of 5014 instances of measured Key Performance Indicators (KPIs) begins to load. The parameters includes 13 features namely, response time, availability, throughput, successability, reliability, compliance, best practice, latency, documentation, Web Service Relevancy function (WSRF), service name and WSDL address as shown in Table 1 with their respective descriptions and statuses. These required parameters were captured for the different webservers (facebook, skype, Youtube, e.t.c.). The data collected goes into the system for processing and the signals generated used as input to the quality of experience predictive system. The features will be generated from the data set in the feature extraction module.

The QoE feature extraction model developed with principal component analysis (PCA) algorithm and implemented with Python scikit – learn library. The procedure of PCA is shown below:

Input: M_Tel QoS dataset
Output: Pre-processed QoS data.
Procedure:
1. Perform QoS data Preprocessing task.
   a. Load dataset which contains default data features
   b. Augment the features with four other features
   c. Import the dataset and save the file in the directory containing the dataset
   d. Extract the independent variables
   For every feature in the dataset
   i. If more than 75 % of the values are missing, then remove a specific row that has a null value for a feature or a particular column.
   ii. If feature value is numeric, calculate the mean for the values in the column and replace with the result for the missing values.
   iii. If the feature is a categorical data, then change the values to numbers since learning models are primarily based on mathematical equations.

The preprocessed data were then stored in QoE server.
The QoE server holds the data when packets are sent on a network. It sends and receives requests from various servers (youtube, facebook, Skype, whattsapp e.t.c) and records them depending on the web site the user wants to visit and then passes the QoE data captured for training.

In model training, features extracted were used as input to train a QoE model. Random forest classifier was used to classify the user's experience into 4 classes which include class 1 (very good QoE), class 2 (good QoE), class 3 (fair QoE) and class 4 (poor QoE) and the performance of the system was evaluated using different accuracy metrics.

3.1 Data set generation for model Building

The dataset used in this work was the Quality of Service (QoS) dataset obtained from M-Tel Nigeria. It originally had thirteen features as shown in Table 1. Feature no. 11 indicates the service classification which takes values from 1 to 4, with the value 1 as the highest quality of service; while the value 4 indicates the lowest quality of service. Feature numbers 12 and 13 were considered unimportant in the prediction of the quality of experience of the websites, and consequently deleted from the dataset.

S/No	**Feature**	**Description**	**Status**
1	Response Time	Time taken to send a request and receive a response	Used
2	Availability	Number of successful invocations/total invocations	Used

3	Throughput	Total Number of invocations for a given period of time	Used
4	Successability	Number of response / number of request messages	Used
5	Reliability	Ratio of the number of error messages to total messages	Used
6	Compliance	The extent to which a WSDL document follows WSDL specification	Used
7	Best Practise	The extent to which a Web service follows WS-I Basic Profile	Used
8	Latency	Time taken for the server to process a given request	Used
9	Documentation	Measure of documentation (i.e. description tags) in WSDL	Used
10	WSRF	Web Service Relevancy Function: a rank for Web Service Quality	Used
11	Service Classification	Describes the level of Quality of Service	Class of QoS
12	Service Name	Name of the Web service	Not used
13	WSDL Address	Location of the Web Service Definition Language (WSDL) file on the Web	Not used

Table 1. Quality of Service Dataset features

4.0. Experiments and results

A series of experiments were carried out by picking random samples of dataset features and the procedures are as follows:

1. Pick random samples from the dataset. Start with four samples for each decision tree.
2. Construct a decision tree for each sample. Each sample with the same node was built using different data that leads to different leaves and get a prediction result from each decision tree. The reason behind constructing a set of decision rules is to align the parameters of a particular sample with its trained values for easy classification and prediction.
3. 50 trees were chosen to form a forest and repeat steps 1 – 2 for 100 and 200 trees. Because it is a classification problem, each tree in the forest predicts the category to which the new record belongs.

4. Vote for each predicted result was perfomed. The new record was assigned to the category that won the majority vote.
5. Select the prediction result with the most votes as the final prediction.

In implementing this model, a couple of tools were used. The Salford Predictive Modeler (SPM) 8.2 was used to analyze the dataset. The variable importance of the features in the dataset were extracted in SPM 8.2 environment; making it possible to identify the features that played a role in the determining the class of an instance. Similarly, the Random Forest algorithms on SPM 8.2 were used to train similar models with their results compared with that of the developed model. Also the QoE data extraction subsystem (which involved the bootstrapping of some features of the real dataset and the multivariate generation of some other features) were implemented in R (R Core Team, 2018), where the features of the datasets were analysed and parameters extracted from the dataset. The ***PCA*** algorithm was implemented in Python 3.6 platform using the Science Kit Learn Library available on the Anaconda 5.2 data science package for windows. The ***QoEPS*** algorithm on the other hand was developed partly in R and partly in Python 3.6.

The Graphical User Interface was developed using the Python Tkinter package, which is a toolkit for the programming of Graphical User Interface. All graphs were plotted in SPM 8.2 package and on Python 3.6 platform using the matplotlib. Similarly, the UML package used in the design of the use case and sequence diagrams was Visual Paradagm (Enterprise version)

Table 2: QoE classification result

Parameter	Value
Number of instances of dataset	364
Number of instances of class 1 (Very good QoS)	41
Number of instances of class 2 (Good QoS)	100
Number of instances of class 3 (Fair QoS)	120
Number of instances of class 4 (Poor QoS)	103
Number of instances of extracted parameters for QoS prediction	5014

Table 3: Accuracy metrics of QoE metrics model against CART algorithm

Test Proportion	Test Parameter	**QoE** Model	CART (Splitting function)	
			Gini	**Information Gain**
0.2	**Accuracy**	92.00	84.35	82.50
	Precision	91.50	86.67	82.25
	Sensitivity	90.50	81.50	84.75
	Specificity	93.50	87.53	80.53

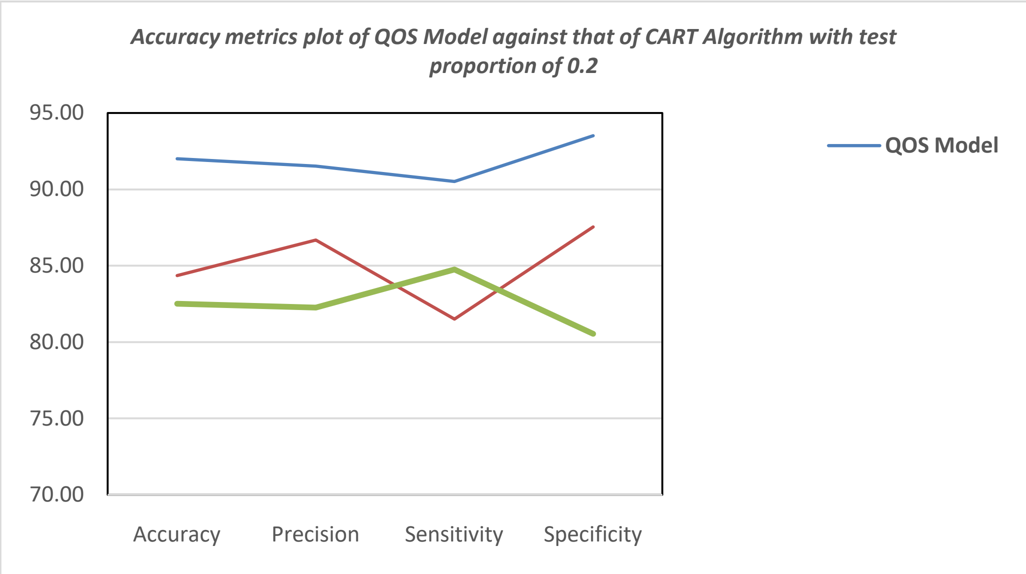

Figure 2: QoE model against CART algorithm

On comparing the results produced by QoE model against those produced by the CART algorithm, the new system showed an improvement on the accuracy metrics over those of CART.

Table 4: Accuracy metrics of the QoE **model against Random Forests Algorithms**

Test Proportion	Test Parameter	QoE Model	Random Forests	
			50 Trees	**100 Trees**
0.4	**Accuracy**	93.50	83.53	84.15
	Precision	92.00	88.16	84.75
	Sensitivity	91.50	84.35	86.15
	Specificity	95.50	86.35	82.43

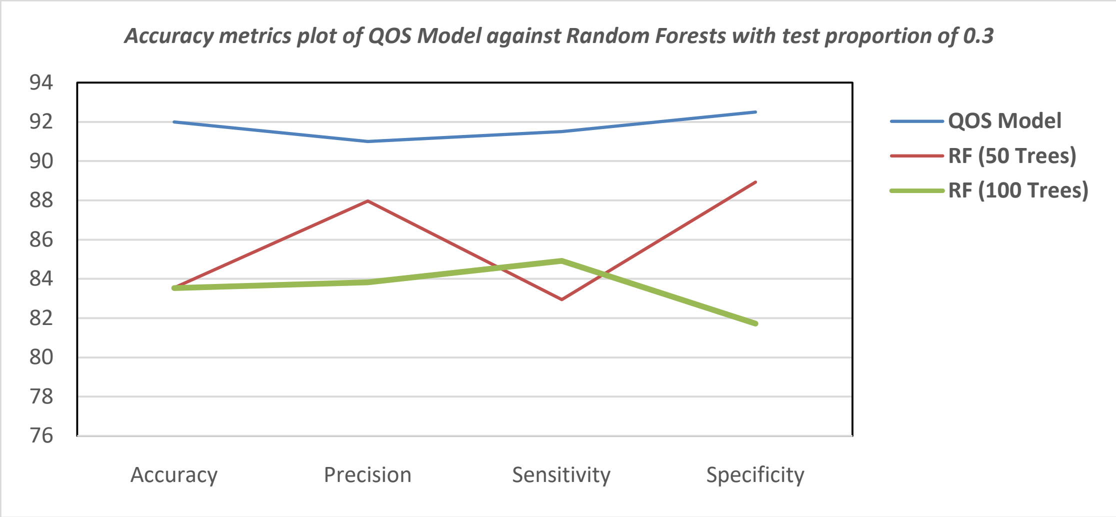

Figure 3: Accuracy plot of QoE **Model against Random Forests Algorithm**

Figures 3 shows the Accuracy plots of the QoE model compared against that of Random Forest models of 50 and 100 trees respectively for test proportions of similar values. It can be seen in the figures that the QoE model consistently outperformed models built on both CART Algorithm and Random Forests Algorithm of varying tree sizes.

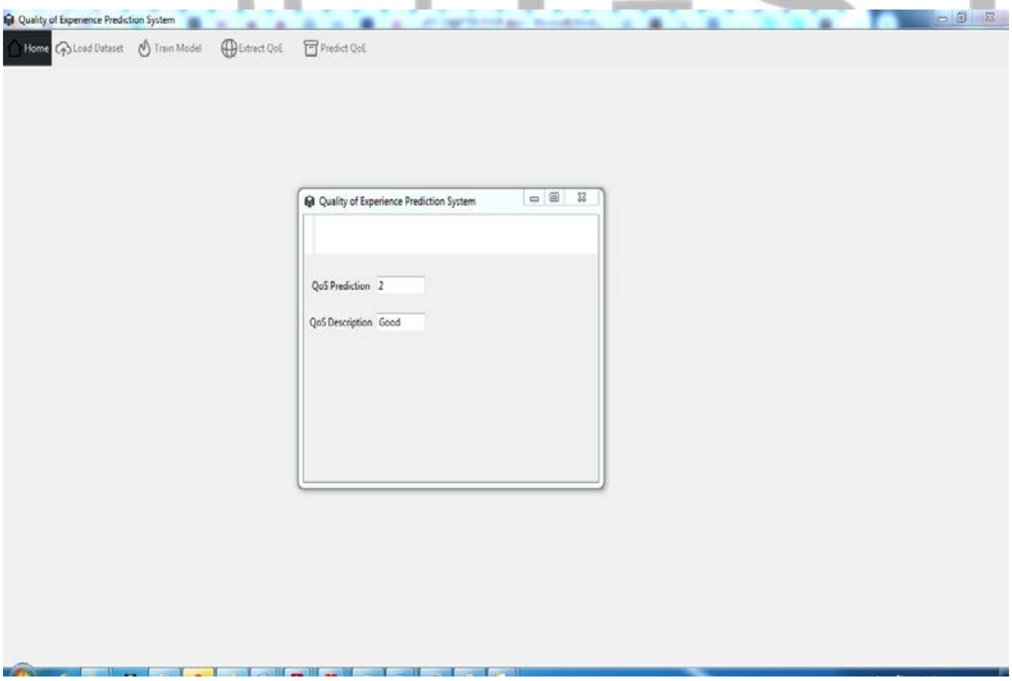

Figure 4: QoE **prediction output for a single website**

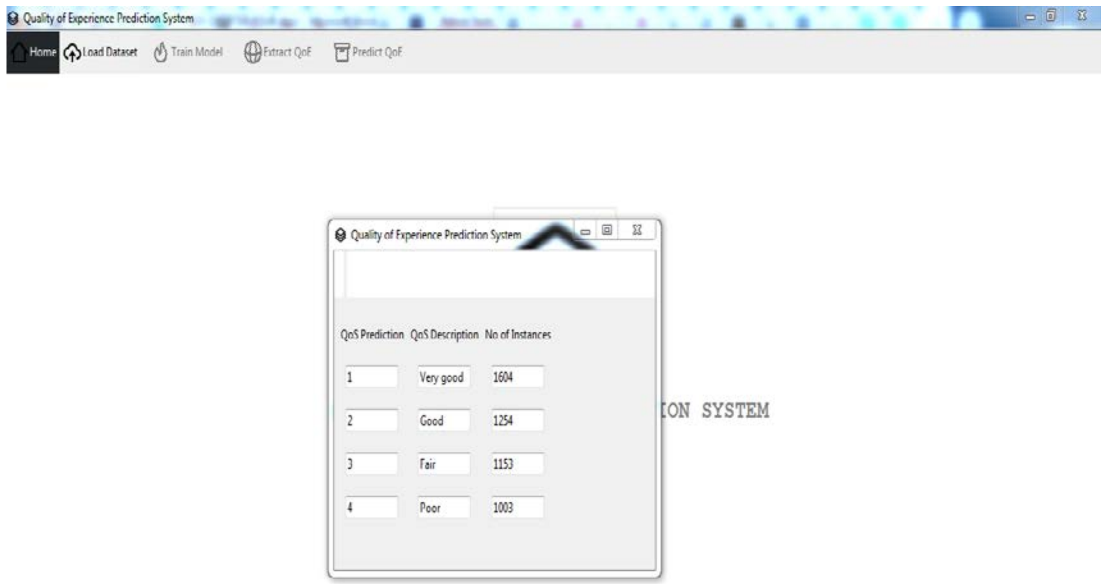

Figure 5: QoE **prediction output for multiple website**

5.0 Discussion of Results

There were a total of 364 predictive results from the model used as shown in table 2. 41 instances of the dataset had a class of 1 (Very good quality of experience). Also, 100 instances of the dataset had a class of 2 (Good quality of experience). There is also a 120 instances of class 3 (Fair quality of experience) and 103 instances of class 4 (Poor quality of experience). Using the network QoS parameter extraction feature which was modeled in the system, 5014 instances of Quality of Service data was captured which was subsequently used in validating the model by predicting the QoE of web users.

Figure 4 shows a sample output of the notification received by the web user from his device predicting to the user that the website he was trying to access would give a good quality of experience. This message is the same for any instance of the experiment where the user is trying to access a live website.

Figure 5 shows the output from the multiple websites where user's key quality indicators were extracted. A total of 5014 websites were analysed and their QoE predicted using the network parameter extraction model of the quality of experience predictive system *(QoEPS)*. In predicting the QoE, 1604 websites were observed to have a QoE class of 1 (very good quality of experience), 1254 websites were also observed to have a QoE of class 2 (Good quality of experience). Similarly, a total of 1153 websites were further predicted to have a QoE class of 3 (Fair quality of experience), while a total of 1003 websites were predicted to have a QoE class of 4 (Poor quality of experience).

5.1 Model Evaluation

The confusion matrix of the QoE model is shown in table 5. The diagonal line in the table depicts the number of predictions that were correctly predicted as the actual classes. Out of the 145 classification results obtained, 128 were correctly predicted as the actual classes: 13 out of 16 were predicted correctly for class 1, 36

out of 40 was predicted correctly for class 2, 43 out of 48 was predicted correctly for class 3 while 36 out of 41 was correctly predicted for class 4. On model evaluation, 12 experiments were carried out with test proportions of 0.2, 0.3 and 0.4. Out of the 145 classification results obtained, 128 were correctly predicted as the actual classes: 13 out of 16 were predicted correctly for class 1, 36 out of 40 were predicted correctly for class 2, 43 out of 48 were predicted correctly for class 3 while 36 out of 41 were correctly predicted for class 4. This gave rise to a prediction result of 93.5 % accuracy, 92 % precision, 91.5 % sensitivity and 95.5 % specificity.

Table 5: The Confusion Matrix of the QoE **Model**

Actual Class	Predicted Class			
	Class 1	**Class 2**	**Class 3**	**Class 4**
Class 1	**13**	**0**	1	2
Class 2	1	**36**	2	1
Class 3	1	1	**43**	3
Class 4	2	1	2	**36**

6. Conclusion

Web quality of experience prediction lacks user centric system which necessitated the study into quality of experience measurement, evaluation and prediction of web users. A predictive model of quality of experience has be developed to help in measuring, monitoring and predicting the experience of web users. The model developed was trained, and evaluated to help in tracing where network service bottle neck could be coming from. This also enables users take useful decisions concerning their web sessions and usage in real time.

7. REFERENCES

Alreshoodi, M., & Woods, J. (2013). Survey on QoE/QoS correlation models for multimedia services. *International Journal of Distributed and Parallel Systems* 4(3), 53– 72. Retrieved from http://arxiv.org/abs/1306.0221 on June 5th 2021

Asrese, A. S., Walelgne, E. A., Bajpai, V., Lutu, A., Alay, Ö., & Ott, J. (2019). Measuring web quality of experience in cellular networks. In *International Conference on Passive and Active Network Measurement*. 18-33. Springer, Cham.

Balachandran, A., Sekar, V., Akella, A., Seshan, S., Stoica, I., & Zhang, H. (2013). Developing a predictive model of quality of experience for internet video. *ACM SIGCOMM Computer Communication Review*, *43*(4), 339- 350.

Baraković, S., & Skorin-Kapov, L. (2017). Survey of research on Quality of Experience modelling for web browsing. *Quality and User Experience*, *2*(1), 1-31.

Boz, E., Finley, B., Oulasvirta, A., Kilkki, K., & Manner, J. (2019). Mobile QoE prediction in the field. *Pervasive and Mobile Computing*, *59*, 101039.

Butkiewicz M., H. V. Madhyastha, and V. Sekar.(2011). Understanding Website Complexity: Measurements, Metrics and Implications. *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. 313-328

Casas, P., D'Alconzo, A., Wamser, F., Seufert, M., Gardlo, B., Schwind, A., & Schatz, R. (2017,). Predicting QoE in cellular networks using machine learning and in-smartphone measurements. In *2017 Ninth International Conference on Quality of Multimedia Experience* 1-6.

Gómez, G., Hortigüela, L., Pérez, Q., Lorca, J., García, R., & Aguayo-Torres, M. C. (2014). YouTube QoE evaluation tool for Android wireless terminals. *EURASIP Journal on Wireless Communications and Networking*, *4*(1), 1-14.

Huang, J., Qian, F., Guo, Y., Zhou, Y., Xu, Q., Mao, Z. M., ... & Spatscheck, O. (2013). An in-depth study of LTE: Effect of network protocol and application behavior on performance. *ACM SIGCOMM Computer Communication Review*, *43*(4), 363-374.

Lycett, M., & Radwan, O. (2019). Developing a quality of experience (QoE) model for web applications. *Information Systems Journal*, *29*(1), 175-199.

Qiao Z,.(2011). Smarter Phone based Live QoE Measurement. In: *15th International Conference on Intelligence in Next Generation Networks* (ICIN

2011), Berlin, Germany.

Reiter, U., Brunnström, K., De Moor, K., Larabi, M. C., Pereira, M., Pinheiro, A., ... & Zgank, A. (2014). Factors influencing quality of experience. In *Quality of experience*. 55-72. Springer, Cham.

Tsolkas, D., Liotou, E., Passas, N., & Merakos, L. (2017). A survey on parametric QoE estimation for popular services. *Journal of network and computer applications*, *7(7)*, 1-17.

Vasilev, V., Leguay, J., Paris, S., Maggi, L., & Debbah, M. (2018). Predicting QoE factors with machine learning. In *2018 IEEE International Conference on Communications (ICC)* 1-6.

Wuruola A.,2018): Framework for Modeling Mobile Network Quality of Experience through Big Data Analytical Approach. *Journal of ICT*, 1, 79- 113.