

Dynamic Resource Allocation in LTE Radio Access Network Using Machine Learning Techniques

Eric Michel Deussom Djomadji^{1,2}, Ivan Basile Kabiena^{2,3}, Valery Nkemeni⁴,
Ayrton Garcia Belinga À Njere², Michael Ekonde Sone¹

¹Department of Electrical and Electronic Engineering, College of Technology, University of Buea, Buea, Cameroon

²Division of ICT, National Advanced School of Post, Telecommunication and ICT, University of Yaounde I, Yaounde, Cameroon

³Department of Computer Engineering and Telecommunications, National Advanced School of Engineering, University of Douala, Douala, Cameroon

⁴Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea, Buea, Cameroon
Email: eric.deussom@gmail.com

How to cite this paper: Djomadji, E.M.D., Kabiena, I.B., Nkemeni, V., Njere, A.G.B.À. and Sone, M.E. (2023) Dynamic Resource Allocation in LTE Radio Access Network Using Machine Learning Techniques. *Journal of Computer and Communications*, 11, 73-93.

<https://doi.org/10.4236/jcc.2023.116005>

Received: May 21, 2023

Accepted: June 27, 2023

Published: June 30, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Current LTE networks are experiencing significant growth in the number of users worldwide. The use of data services for online browsing, e-learning, on-line meetings and initiatives such as smart cities means that subscribers stay connected for long periods, thereby saturating a number of signalling resources. One of such resources is the Radio Resource Connected (RRC) parameter, which is allocated to eNodeBs with the aim of limiting the number of connected simultaneously in the network. The fixed allocation of this parameter means that, depending on the traffic at different times of the day and the geographical position, some eNodeBs are saturated with RRC resources (overused) while others have unused RRC resources. However, as these resources are limited, there is the problem of their underutilization (non-optimal utilization of resources at the eNodeB level) due to static allocation (manual configuration of resources). The objective of this paper is to design an efficient machine learning model that will take as input some key performance indices (KPIs) like traffic data, RRC, simultaneous users, etc., for each eNodeB per hour and per day and accurately predict the number of needed RRC resources that will be dynamically allocated to them in order to avoid traffic and financial losses to the mobile network operator. To reach this target, three machine learning algorithms have been studied namely: linear regression, convolutional neural networks and long short-term memory (LSTM) to train three models and evaluate them. The model trained with the LSTM algorithm gave the best performance with 97% accuracy and was therefore implemented in the proposed solution for RRC resource allocation.

An interconnection architecture is also proposed to embed the proposed solution into the Operation and maintenance network of a mobile network operator. In this way, the proposed solution can contribute to developing and expanding the concept of Self Organizing Network (SON) used in 4G and 5G networks.

Keywords

RRC Resources, 4G Network, Linear Regression, Convolutional Neural Networks, Long Short-Term Memory, Precision

1. Introduction

Since the emergence of mobile telephony, the generations of networks have continued to evolve in order to offer an ever more satisfactory quality of service. The radio technology used aims at sharing a frequency spectrum in several frequencies between multiple users, distributed in different radio cells according to their spatial situation. Thus, to communicate from a cell phone, it is necessary to be within range of a Base Transceiver Station installed by the mobile operator in order to receive a sufficiently strong radio signal. Over the years, the global demand for mobile data services has grown phenomenally, generating a heavy network signaling load. This consumes a disproportionate amount of network resources, compromising network throughput, efficiency, and quality of service. Many ideas are proposed to solve the problem of exploding mobile data traffic, including the small cell approach [1] and D2D communication [2], etc. However, it can be noted that such efforts focus primarily on how to manage radio resources. Several generations of networks have been developed, each time improving performance to meet user requirements.

It can also be noted that, despite the presence of a 5th network generation (5G) with more advanced technology and higher performance in terms of throughput, the 4th generation remains the dominant technology in the majority of countries in the world today. Indeed, LTE technology still dominates global mobile telecommunications. There were 5.95 billion LTE subscriptions globally at the end of the third quarter of 2020, and LTE technology accounts for approximately 62.1% of mobile subscriptions globally [3].

The generation of complex and large amounts of data by the networks of telecommunications operators has motivated us to seek a machine learning-based technique that can be used to overcome the present challenges of resource allocation experienced by these networks.

A mobile network operator (MNO) whose LTE access network (E-UTRAN) infrastructure is supervised by the network management system U2020 platform was used as a case study in this work.

The U2020 has several management functions including topology management, alarms management, software management, configuration management,

and license management. From the license management function, resources can be allocated to eNodeB based on the total resources purchased by the MNO in a fixed manner. This implies that the network planning engineer based on current resources utilization will decide the number of resources that will be assigned to each eNodeB. This is the case with Radio Resource Connected (RRC) allocation. Each eNodeB will be configured with an adequate number of RRC (Radio Resource Connected) resources needed to ensure the admission requests of users in each of the cells it manages. The network will work as configured, but it may happen that at certain periods, some eNodeBs do not have enough resources to continue to satisfy the users' requests. This limitation is the main motivation for this study, which involves the implementation of an efficient machine learning model that will take as input the eNodeBs' traffic data and will accurately predict the number of RRCs to be allocated to each eNodeB on an hourly basis to ensure a dynamic allocation of resources that will minimize the observed traffic losses. Artificial intelligence methods have been selected because they have great processing and adaptation capabilities.

In fact for a Mobile network operator running a 4G LTE network, based on the purchased licenses, a radio network planning engineer will assign resources to each eNodeB to operate. This resource is a permanent resource that belongs to the eNodeB. But the problem faced is that during the day, the traffic per eNodeB is dynamic and will depend on the hour of the day and the position of the eNodeB in the town. For example, during working days of the week, eNodeBs in the center town will have a high traffic from 8:00 AM to 5:00 PM, thus requiring high RRC resource utilization while at the same time, eNodeBs in the suburbs have small traffic demands and the RRC utilization is low. This can be explained by the fact that people leave their houses in the morning, go to work mainly in the centre town and come back to their homes in the evening after work. Also, from 6:00 PM to 11:00 PM, it is noticed that the traffic in suburbs becomes very high while traffic in centre town reduces; this is the same for the RRC resources utilization. Because of the fixed allocation of resources especially in the case of RRC, when the maximum number of RRC per site is reached, users can no more register to the eNodeB to enjoy services. This is a kind of service rejection that will lead to both traffic and financial loss because users have their bundle available, but they cannot use it when they want due to congestion and the fact that RRC licenses are exhausted in some eNodeBs. At the time when some eNodeBs have their RRC resources completely used, some eNodeBs have the RRC available and not used. This limitation resulting from fixed resource allocation is the problem this study intends to solve. The question that needs answers is how machine learning techniques can be used for dynamic resource allocation based on traffic and RRC resources utilization. **Figure 1** presents a case of a fixed RRC resources allocation where eNodeB_1 has 100 RRC assigned, eNodeB_2 has 200 RRC, eNodeB_3 has 100 RRC and eNodeB_N has 150 RRC for a total of M RRC purchased by the MNO. This means that the RRC is a limited resource at the level of the MNO that needs optimal allocation.

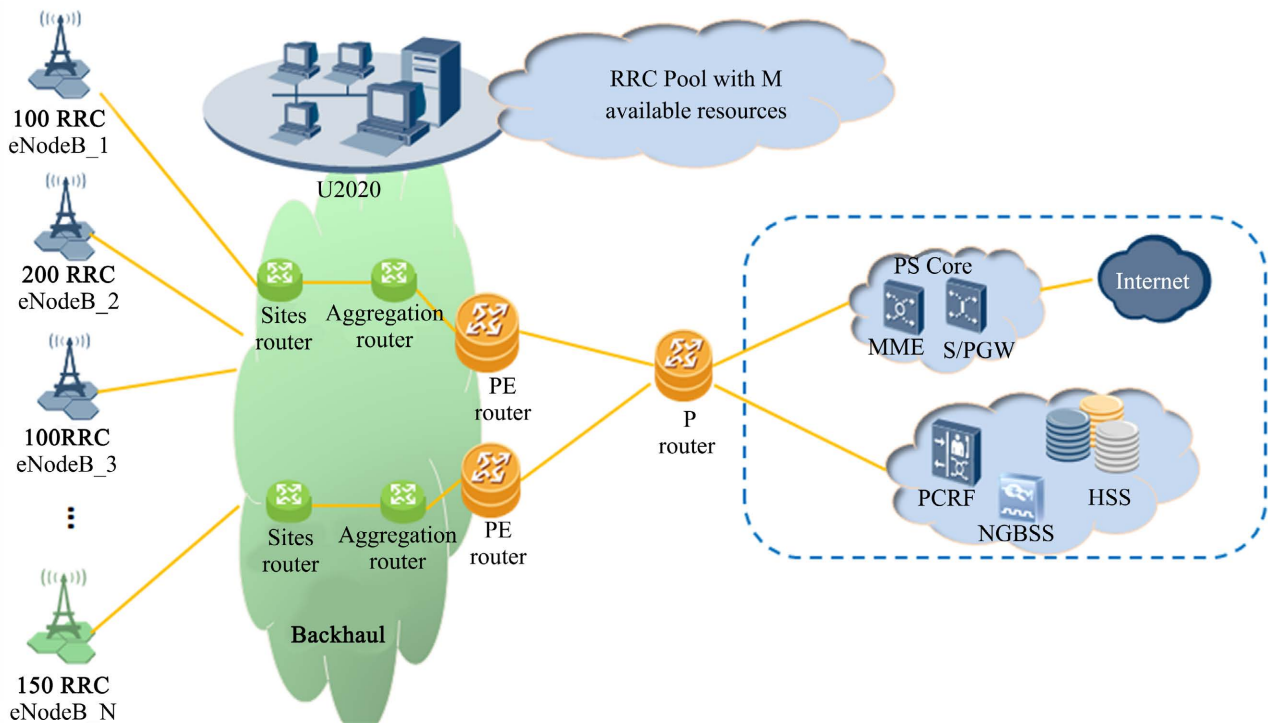


Figure 1. LTE network with a case of fixed RRC resource assignment.

From the problem depicted in **Figure 1**, this study is interested in developing a machine learning-based solution that dynamically allocates RRC resources to eNodeBs on an hourly basis, depending on the utilization experience. This paper starts with the introduction, followed by the presentation of the existing methods for resource allocation in the second section, and materials and methods in the third section. The fourth section describes the experiments conducted and the results obtained follow by the results discussion. Lastly, the conclusion and perspectives are presented.

2. Existing Methods

Dynamic resource allocation in telecommunications networks is a problem that has been addressed by several researchers in the world over the years via different methods. This problem arises at several levels in a network, with an example being the air interface between the eNodeB and mobiles. For our case study, our focus is to address this problem on the interface between NMS and eNodeBs (between the U200 and the eNodeBs for our case study).

In the air interface, Bahreyni and Sattari-naeini in 2014 [4] proposed a new scheduling algorithm for LTE networks that supports rapid channel variations and aims to increase system capacity while maintaining fairness when the number of active users is greater than the number of existing RBs. This algorithm improved the performance of cell edge users based on the preferences of users with less bandwidth. During the simulation, the proposed algorithm was compared with Round Robin (RR), Best CQI and Proportional Fair (PF). The results

showed a good level of fairness with little decrease in user throughput and total system throughput, indicating that this algorithm is dedicated to ensuring a good level of fairness among users, even if the minimum QoS level is reached.

Habaebi *et al.* [5], presented a comparison of three scheduling algorithms namely: RR (Round Robin), best CQI, and PF (Proportional Fair) schedulers. The performance of the downlink scheduling algorithms was measured in terms of throughput and block error rate (BER) using a MATLAB-based system-level simulator. The results indicated that the Best CQI algorithm performed better than the others in terms of throughput levels, but at the expense of fairness to other users with poor channel conditions.

LTE-A introduces important features such as coordination between eNBs (LTE-A base station), cognitive radio, capacity, and coverage improvement. A cross-layer resource allocation method for inter-cell interference coordination (ICIC) in LTE-A is analyzed in [6]. The inter-NB coordination for ICIC uses evolutionary game theory to avoid the interference of cell resource allocation strategies. Particle swarm optimization (PSO) is used to find the best scheduling scheme for each resource block in the multicell case. The results are compared with RR, PF, max C/I and MLWDF scheduling schemes. The cross-resource allocation improves the system throughput remarkably and fairness is guaranteed. The time complexity of this new approach is determined by the price of the anarchic solution. With this parameter, the cross-layer scheme is based on potential sets and the PSO algorithm finds the optimal solution only after 2 or 3 iterations.

Many authors have used machine learning to solve problems in the field of telecommunication and mobile networks. Jean MELI TAMWA *et al.* in [7] worked on “Intrusion Detection aided by Artificial Intelligence”. DEUSSOM Eric *et al.* in [8] worked on “Machine learning-based approach for designing and implementing a collaborative fraud detection model through Call data records CDR and traffic analysis”, in their work they used machine learning methods to detect and identify frauds on mobile network internet bundles by analyzing CDR and traffic. Deussom Djomadji Eric *et al.* in [9] carried out a study on “Machine Learning-Based Approach for Identification of SIM Box Bypass Fraud in a Telecom Network Based on CDR Analysis: Case of a Fixed and Mobile Operator in Cameroon”, in this work they focused on voice call and used machine learning to identify SIM box bypass which has a big impact on mobile network revenue on the international calls segment. Batchakui Bernabe *et al.* in [10] compared machine learning Algorithms for Improving the Maintenance of LTE Networks Based on Alarms Analysis. In another study, Deussom Djomadji Eric *et al.* in [11] used machine learning to perform alarm classification and correlation in an SDH/WDM Optical Network to Improve Network Maintenance. Based on their work, it became easy for engineers working in the SDH/WDM transport network to easily identify the root cause of problems and reduce the downtime of service. This helps to improve the service availability and fulfill the service level agreement between the company and its customers.

3. Materials and Methods

The traffic data to be exploited in this work are labelled data, *i.e.*, the output sought to be predicted is known. Thus, first it is sought to predict the number of RRC resources to allocate to each eNodeB and then an allocation is done. Next, the algorithms used to train the three machine learning models for resource prediction and allocation are presented, namely linear regression, convolutional neural networks (CNN), and long-term short-term memory (LSTM) [12].

3.1. Linear Regression

The most basic model is linear regression [13] where a variable X is explained, and modelled by an affine function of another variable y . The purpose of such a model is multiple and therefore depends on the context and especially on the underlying questions. A distinction is made between simple linear regression, in which only one independent/predictive variable (X) is used to model the response variable (Y); but there may be various cases in which the response variable is affected by more than one predictor variable; in such cases, the multiple linear regression algorithm is used. Moreover, multiple linear regression is an extension of simple linear regression because more than one predictor variable is needed to predict the response variable.

3.2. Convolutional Neural Networks

Convolutional neural networks [14] are based on convolution filters (digital matrices). The filters are applied to the inputs before they are transmitted to the neurons. They have found applications in different domains: facial recognition, text scanning, natural language processing.

3.3. Long Term Short Term Memories (LSTM)

Recurrent neural networks save the results produced by the processing nodes and feed the model with these results. This mode of learning is somewhat more complex. Due to the ability of sequential and temporal modelling, the recurrent neural network (RNN) has been successfully applied to many challenging practices, such as natural language processing, traffic forecasting and crowd density prediction. However, the traditional RNN sometimes suffers from the issue of fading and blow-up gradient because the RNN can no longer be trained properly and inevitably loses performance. To effectively tune these parameters from the first layers would require a lot of time and costly computing resources. With this in mind, a variant of RNN, called the LSTM (Long short-term memory) model, has been designed, which uses memory cells with different gates to store information useful for long-term dependencies [15].

3.4. Environment and Tools Used

To train our different models based on the three machine learning algorithms

previously mentioned, Python programming language was used, in particular version 3.10. It is the reference language used in the development of applications for artificial intelligence. It is easy to install, uncompiled, fast and light. To perform data pre-processing, matrix calculation, and data visualization, libraries such as Pandas, Numpy, Matplotlib were used. For model training, Tensorflow and Keras libraries were used. Finally, google Colab environment was used to develop the different models.

3.5. Dataset Processing

To work with machine learning projects, large amounts of data are needed, because without the data, it is not possible to train ML/AI models. The proposed model analyzes KPIs data from the operation of eNodeBs in the operator's network. For this study, two KPI files were collected, one with 91,719 rows and the other with 83,478 rows. These two files constitute what is commonly called dataset. The next step is to prepare the data, *i.e.* to understand them, to analyze them and to select those which will be used for the construction of the various models. In **Figure 2** some fields of our dataset are presented.

We continue the processing of the data by feature selection which consists of preparing the raw data and fitting it to a machine learning model. The procedure used is presented as follows:

- Delete the rows with NaN values in our dataset with the drop function.
- Then selected the columns highlighting the use of resources by the subscribers and the consumption of data by them. To do this, all the columns that will not be useful for training the model using the Python drop function were deleted, available with the pandas library.
- Then, the two datasets were merged to obtain our 12 features. This is presented in **Figure 3**.
- Finally, it was obtained a usable file of 12 columns and 75,706 lines to train the model.

#4G/LTE DL Traffic Volume (GBytes)(GB)	#4G/LTE UL Traffic Volume (GBytes)(GB)	#4G Data Volume(Gbit)	#4G/LTE DL Traffic Volume (GBytes)(GB)2	#Ave 4G/LTE DL User Thrput (All) (kbps)(Kbps)
0.0001	0.0006	0.0008	0.0001	1448
17.2696	0.9669	19.5813	17.2696	16010.2121
1.0381	0.1257	1.2497	1.0381	32346.1427
11.1897	1.721	13.8627	11.1897	17668.4933
10.3341	8.0191	19.7066	10.3341	20558.2013
8.3216	1.2339	10.2601	8.3216	17083.0143
5.8593	0.2428	6.552	5.8593	14960.1273
0.0851	0.0173	0.1099	0.0851	16087.425
23.0416	2.4146	27.3335	23.0416	17944.6607
20.9385	1.7941	24.409	20.9385	9785.1924
8.1414	0.6332	9.4217	8.1414	15107.0029
14.0393	2.9927	18.2879	14.0393	9846.9326
29.6237	3.6011	35.6749	29.6237	6106.6588
4.0399	0.1657	4.5157	4.0399	23577.6673
12.1783	0.5943	13.7145	12.1783	24228.5994
6.6979	0.2748	7.4869	6.6979	11529.1172
7.4591	0.4296	8.4705	7.4591	14408.7877
6.2843	0.6531	7.449	6.2843	13910.85
4.4243	0.7374	5.5424	4.4243	26666.4693
12.7213	0.6961	14.4068	12.7213	15763.8172
18.65	4.1021	24.4299	18.65	10835.6682

Figure 2. Some 4G KPI's.

1	Time,#4G/LTE DL Traffic Volume (GBytes)(GB),Cell Traffic Volume_DL(Gbit),PS Traffic (kbit),Traffic Volume(GB),L.ChMeas.PR.B.DL.Used.Avg,L.Traffic.User.Max								
2	2022-07-02 00:00:00,17.2696,148.3446,148344619.864,106,36.583,114								
3	2022-07-02 00:00:00,11.1897,96.1188,96118789.68,88,18.0602,109								
4	2022-07-02 00:00:00,10.3341,88.7688,88768833.624,95,8.8115,122								
5	2022-07-02 00:00:00,8.3216,71.482,71482033.904,64,11.8711,77								
6	2022-07-02 00:00:00,5.8593,50.3307,50330666.496,49,6.9581,77								
7	2022-07-02 00:00:00,0.0851,0.7311,731100.56,11,0.9506,26								
8	2022-07-02 00:00:00,23.0416,197.9262,197926170.896,216,24.3618,239								
9	2022-07-02 00:00:00,14.109,121.1957,121195744.896,157,18.7574,182								
10	2022-07-02 00:00:00,6.9571,59.7608,59760816.888,74,14.8658,90								
11	2022-07-02 00:00:00,15.2817,131.2684,131268398.112,101,12.9275,135								
12	2022-07-02 00:00:00,5.0034,42.9786,42978578.536,51,14.9088,59								
13	2022-07-02 00:00:00,9.0011,77.3186,77318636.056,94,5.7962,114								
14	2022-07-02 00:00:00,32.0966,275.7074,275707423.84,210,52.1582,231								
15	2022-07-02 00:00:00,10.2349,87.917,87917049.712,95,15.7065,114								
16	2022-07-02 00:00:00,2.7098,23.277,23276954.96,50,27.5968,54								
17	2022-07-02 00:00:00,22.9272,196.9427,196942722.072,112,17.4771,127								
18	2022-07-02 00:00:00,13.7209,117.8614,117861373.552,87,15.2839,107								
19	2022-07-02 00:00:00,8.6318,74.1464,74146371.088,98,33.7338,111								
20	2022-07-02 00:00:00,3.0303,26.03,26029974.872,49,10.4699,70								Activer

Figure 3. Final data set.

3.6. Model Construction and Training

✚ Separation of the data set

After clearing the data, the next step is to divide our dataset into 3 groups: 70% of the dataset for training the model, 20% for validating the model during training and 10% for testing and validating it if it performs well.

✚ Data normalization

It consists in putting the independent variables of the data set in a specific range. The variables were put in the same range and on the same scale so that no variable dominates the other variable. This is done as shown in Figure 4 below:

✚ Search for dependencies between data

Here it was investigated whether the **user number** (LTE Traffic User Max) which is relative to the number of RRCs that are to be predicted depends on parameters such as traffic volume, traffic volume per cell, number of PRBs, volume of data consumed in downlink, packet data traffic. Figure 5 presents the dependencies between variables.

It can be seen that the target variable (**user number**) is strongly dependent on the downlink traffic volume, the total traffic volume, the packet data traffic and the traffic volume per cell. It can be therefore confirmed that as soon as an accurate model is obtained, the prediction results will also be real because the target is strongly dependent on traffic data.

4. Results

4.1. Model Evaluation

The evaluation aims to verify the model(s) or knowledge obtained to ensure that they meet the objectives formulated at the beginning of the process. It also contributes to the decision to deploy the model or, if necessary, to improve it. At this stage, the robustness and accuracy of the models obtained are tested.


```
[ ] #Normalisation des données d'entraînement, de validation et de test
train_mean = train_df.mean()
train_std = train_df.std()

train_df = (train_df - train_mean) / train_std
val_df = (val_df - train_mean) / train_std
test_df = (test_df - train_mean) / train_std
```

Figure 4. Data normalization.

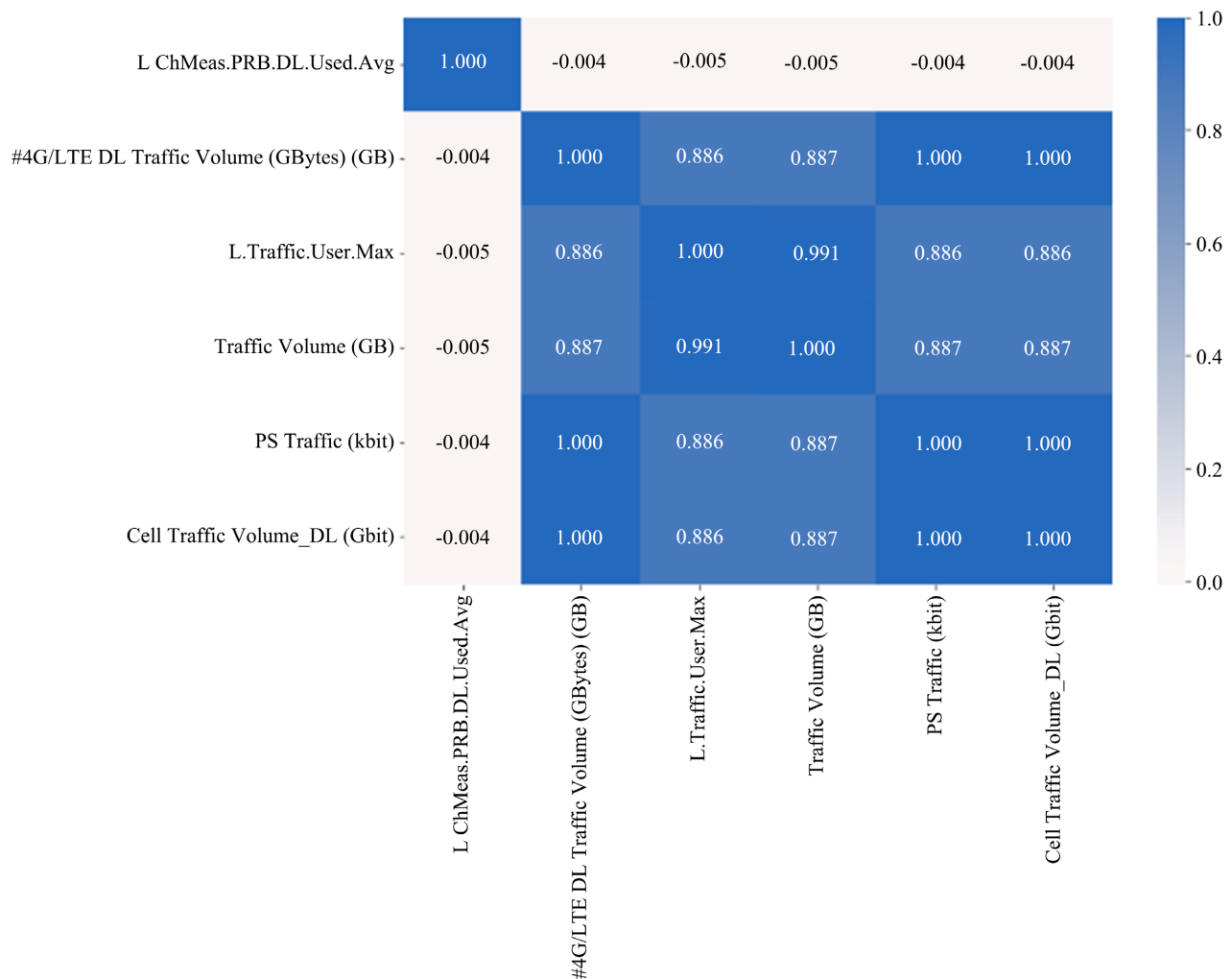


Figure 5. Search for dependencies between variables.

- **The loss curve:** Provides information on the robustness of our model.

The valloss is 1.052 for the linear model, 0.72 for the CNN model and 0.018 for the LSTM model. The best result is thus obtained with the LSTM because the val_loss value tends more towards 0, which means that there is a good learning on the data. The loss curves are presented above in **Figures 6-8**.

- **The test training curves:** Play the same role as the loss curve but they give information about the accuracy of the model.

The prediction points are presented in **Figures 9-11**.

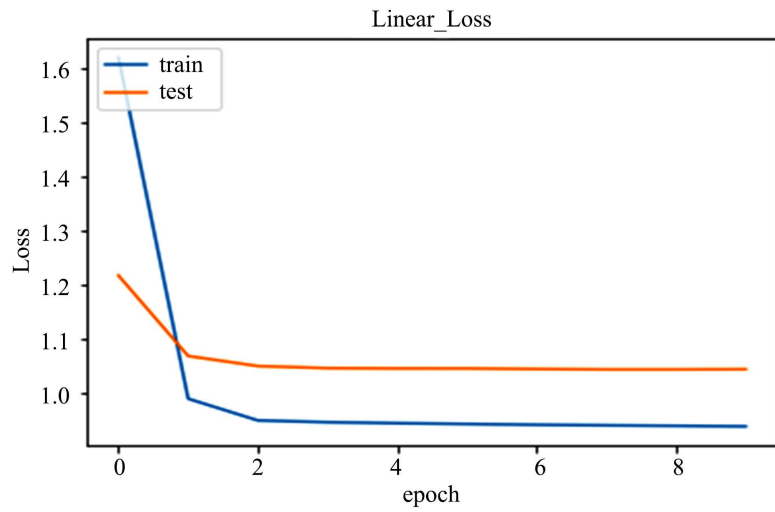


Figure 6. Loss curve was obtained from the linear model.

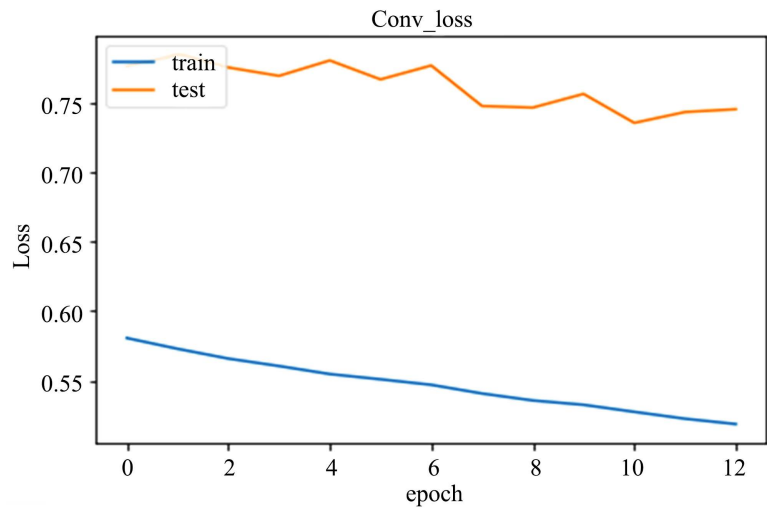


Figure 7. Loss curve was obtained from the CNN model.

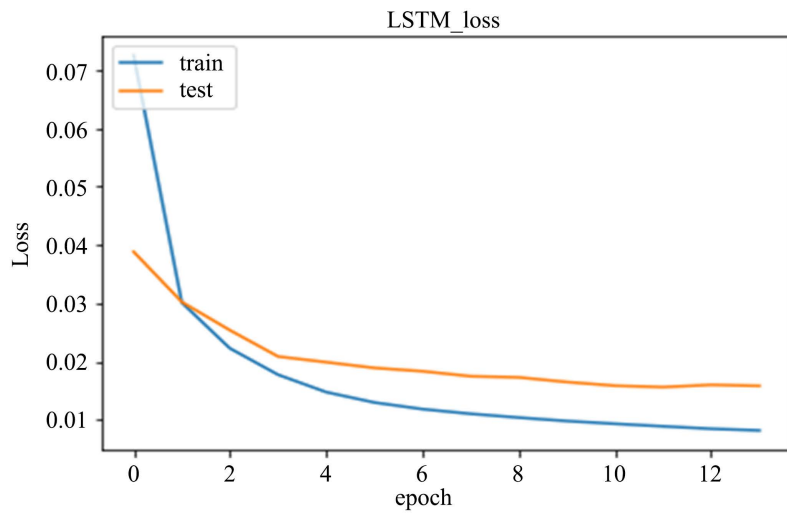


Figure 8. Loss curve obtained with the LSTM.

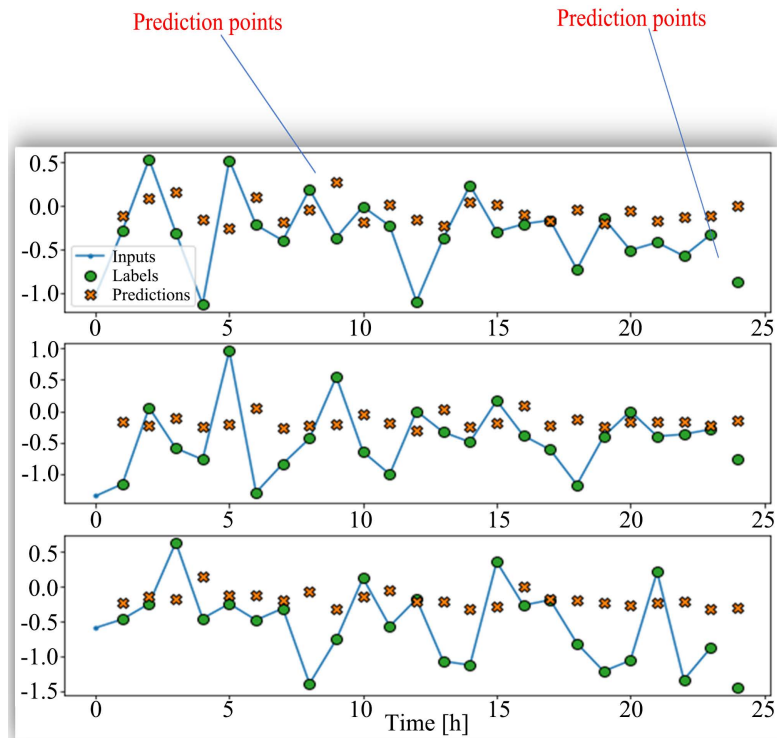


Figure 9. Test training curve on the LTE Traffic User Max with the linear model.

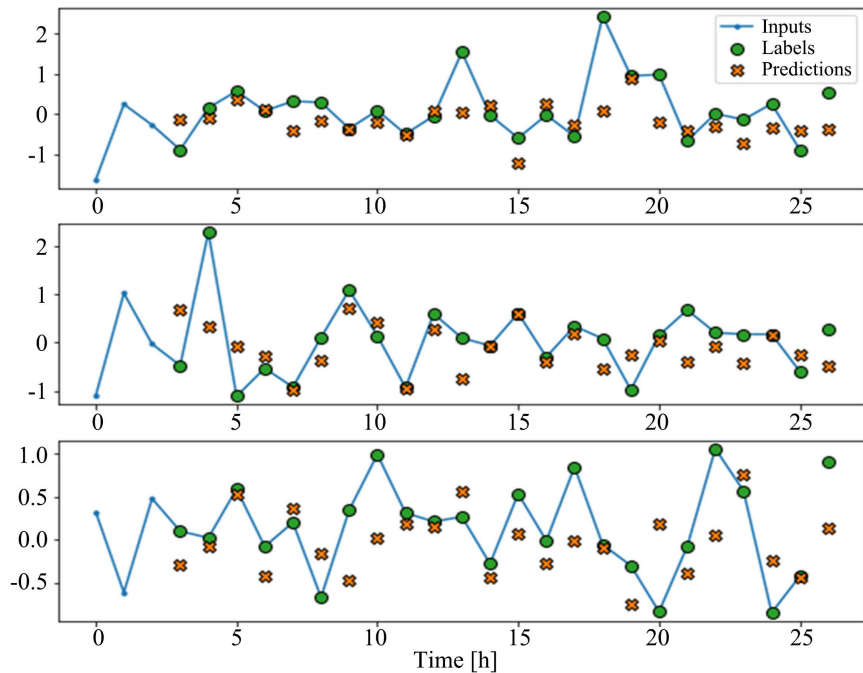


Figure 10. Test training curve on the LTE Traffic User Max with the CNN model.

It was realized that only the test curve with the Long Term Short Term Memories (LSTM) model provides the best accuracy because the prediction points are more similar to the actual value points, so it seems that the LSTM model will give more accurate results than the others.

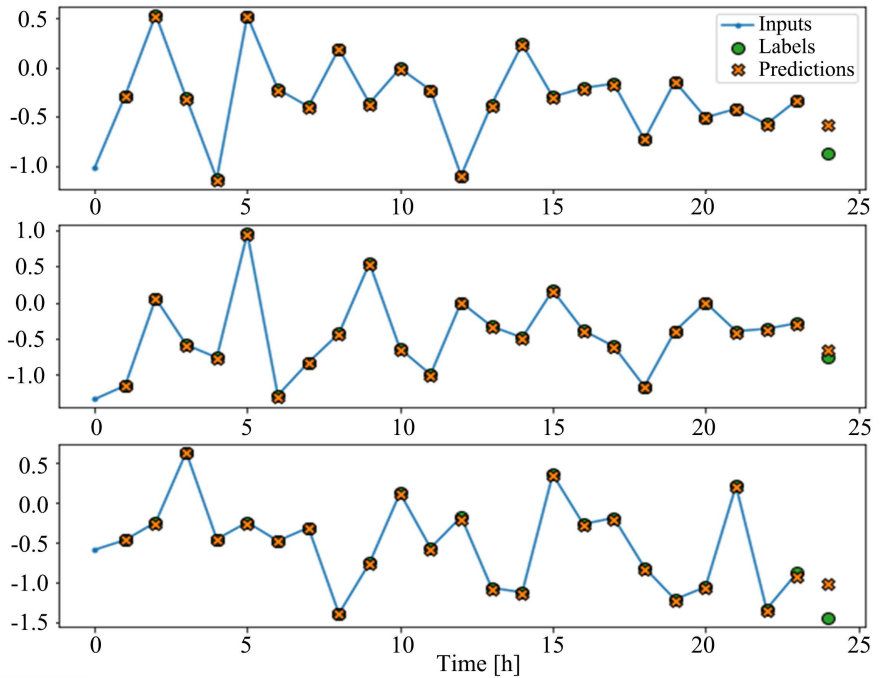


Figure 11. Test training curve on the LTE Traffic User Max with the LSTM model.

➤ **The (MAE) or mean absolute error:** is the mean absolute difference between the model-fitted values and the observed historical data.

The MAE is presented in **Figures 12-14** for each model.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_n - x| \tag{1}$$

By visualizing the three previous graphs translating the MAE, it was realized that the LSTM model gives the lowest error with a value of 0.0285 obtained during the training. This performance is due to the fact that a huge amount (75,000) of data were used to train this model, the other models do not fit very large amounts of data during training.

➤ **Calculation of the accuracy of the different models**

$$PRECISION = 100 - MAE * 100 \tag{2}$$

The model precision is given in **Figure 15** for the 03 models studied in this work.

4.2. Model Validation

In machine learning, model validation is the process by which a trained model is evaluated with a test data set. The main purpose of using the test dataset is to test the generalizability of a trained model. After testing the proposed model with new data, the results depicted in **Figure 20** were obtained.

The results presented in **Figure 16** are predictions that were made over twenty days from our test dataset. Therefore, it is possible to predict the number of resources needed to be allocated to the eNodeBs over a desired number of days.

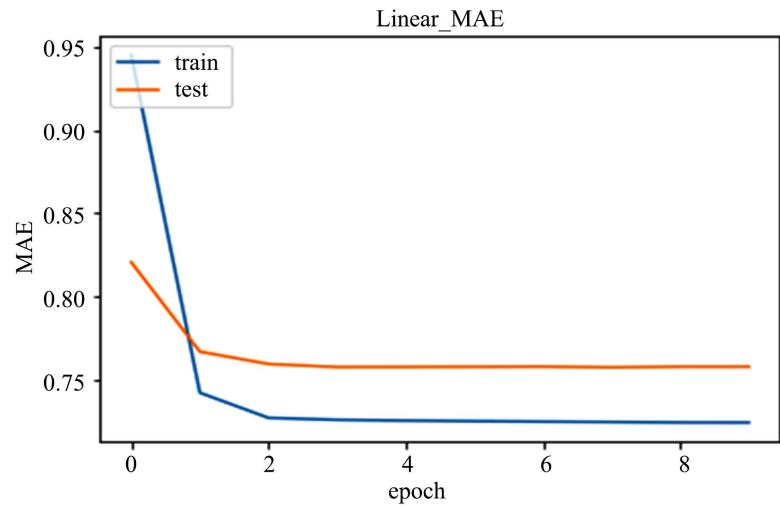


Figure 12. MAE linear model.

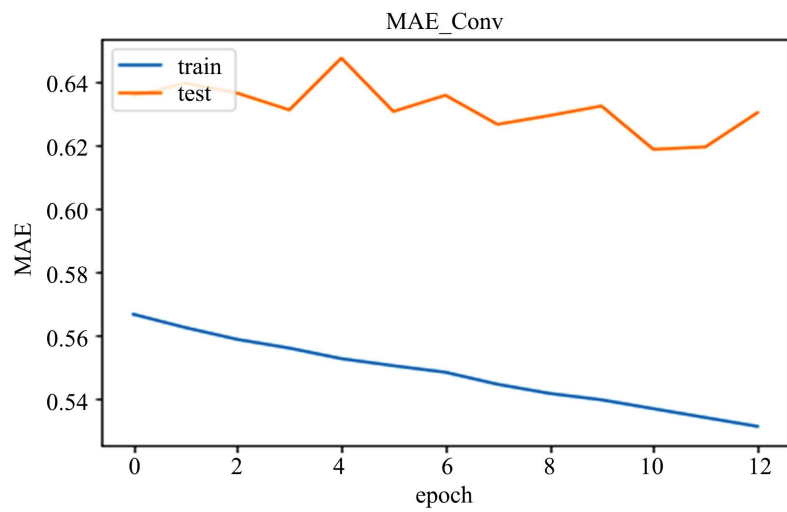


Figure 13. MAE convolutional model.

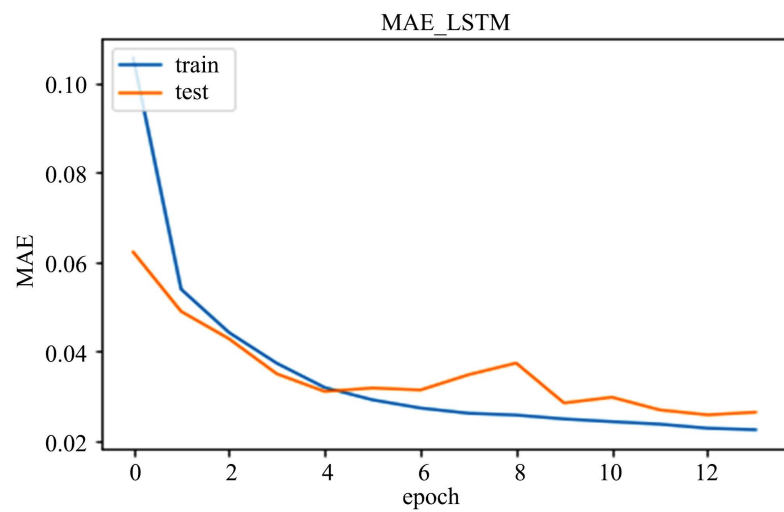


Figure 14. MAE LSTM model.

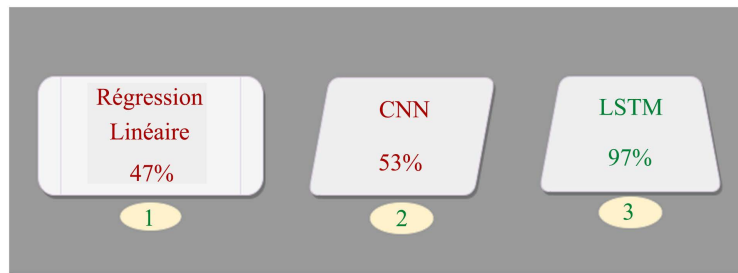


Figure 15. Model accuracy.

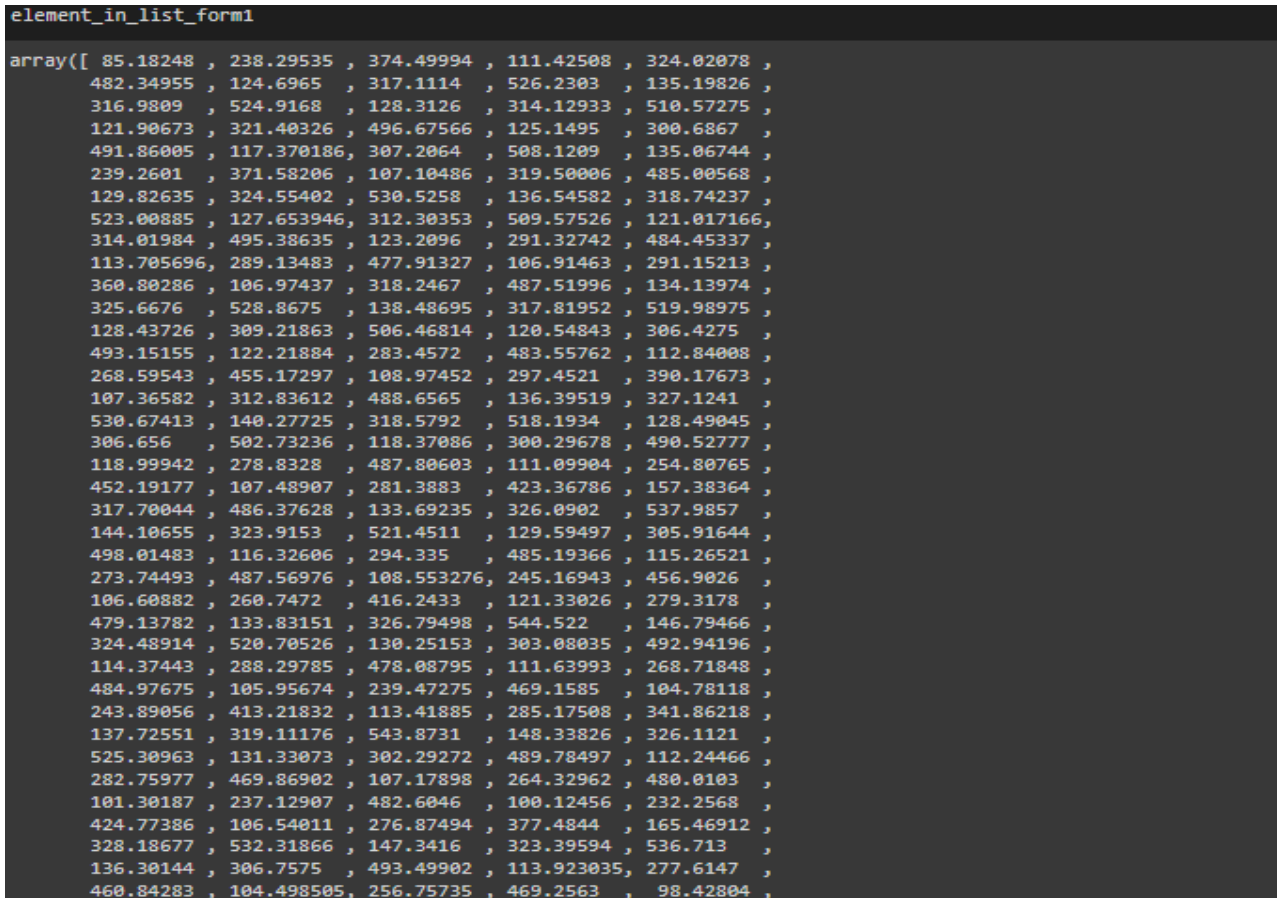


Figure 16. Values predicted by the model.

4.3. Discussion about the Model Validation

At the end of the evaluation of the three models proposed in this work, and considering that there is almost no work done in terms of dynamic allocation of resources to eNodeBs, it was decided to discuss the performances obtained with each of the trained models. Thus, it was obtained respectively with the linear, convolutional and LSTM models accuracies of 47%, 53% and 97%. It should also be noted that these models were trained with real data from the operation of a mobile network operator in Cameroon. Nevertheless, the difference in the results obtained was due to the quantity of data (75,000 lines) used for training, which allowed the LSTM model to win over the others.

4.4. Proposed Solution Deployment and Test

After the identification of the best model to be used for the dynamic resource allocation, it is then possible to propose a deployment architecture of the solution in the network of a mobile network operator and also develop a software or a tool that can automatically perform the data importation, data processing and resource allocation to be transferred to the operation and management platform which in our case study is the U2020, which a NMS for mobile network operators that use Huawei equipment.

✚ Solution architecture

In case we have to implement this solution in a mobile operator's network, the functional architecture would consist of three large interconnected blocks to predict and dynamically allocate the resources necessary to take into account the admission requests of UEs in the LTE cells of the mobile network operator by the eNodeBs in order to ensure better QoS in its 4G network. **Figure 17** presents the architecture of the solution. In this solution a Machine learning algorithm collects data from the U2020 (OMC server), then do all the required calculations to finally proposed a new allocation scheme of RRC to eNodeBs every hour.

The process for dynamically allocating RRC resources using machine learning is presented in **Figure 18**.

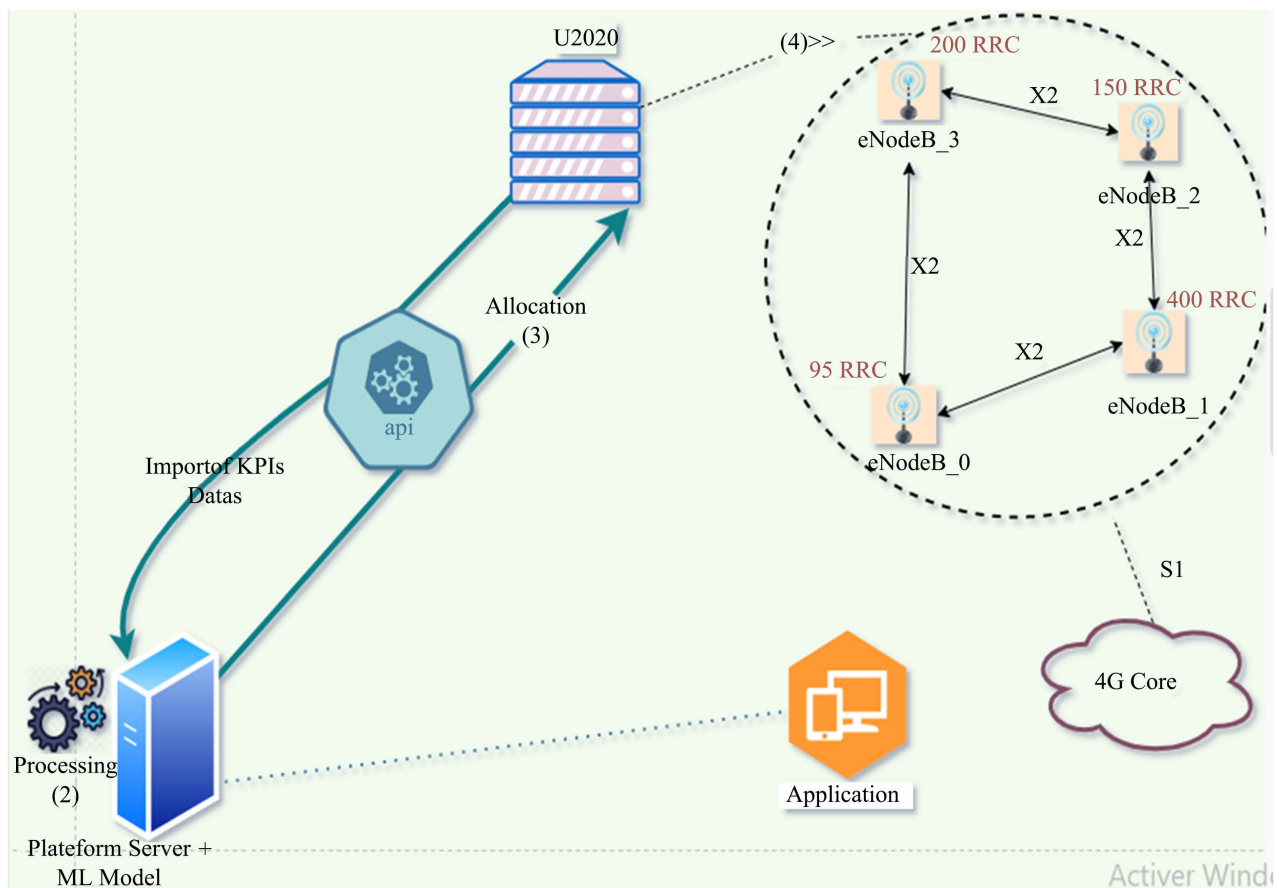


Figure 17. RRC resource allocation implementation method.

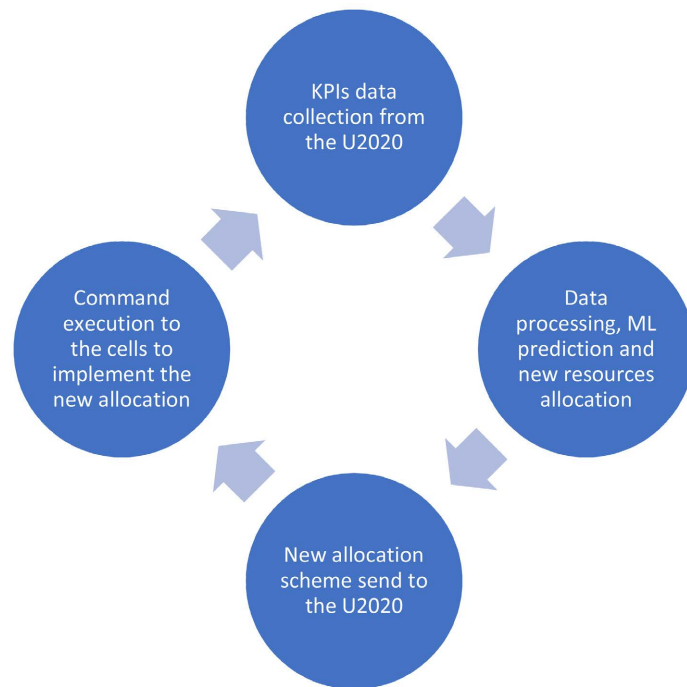


Figure 18. Resources allocation procedure.

The steps involved are as follows:

- 1) KPIs data collection from the U2020 by the proposed Machine learning platform;
- 2) Data processing, prediction and proposal of a new resource allocation;
- 3) New allocation scheme send to the U2020;
- 4) Command execution to the cells to implement the new allocation;
- 5) KPIs data collection from U2020 for the next cycle (every hour for example).

The following explanations can be added also to well understand the function of each block of **Figure 17**.

- The 1st block represents the server which contains the already trained model which will be responsible for collecting the KPIs data from the U2020 API and making predictions every hour and making the dynamic allocation every hour in case the data of traffic changes;
- The 2nd block is the U2020 which contains all KPIs including the RRC Connected resources that should be allocated to the eNodeB and which provide the interface for commands configuration to do the resource allocation itself to the cells;
- The 3rd block represents the eNodeBs which will receive the resources allocation scheme every hour to ensure the admissions of the users in the LTE cells and avoid admission rejection due to RRC resources congestion.

4.5. Presentation of RRC Prediction Tool Developed

At the end of this work, a web-based software was developed to implement the

proposed solution for dynamic resource allocation of RRC resources. Though this work focused on RRC resource allocation, it can be easily generalized for other LTE radio resources allocations, in fact for user equipment to be registered in an LTE cell, an RRC should be available and allocated to it, otherwise, it cannot access nor enjoy the LTE services.

Figure 19 presents the login interface of the proposed software and figure presents the dashboard of the software.

The dashboard presented in **Figure 20** here shows the various operating parameters such as the number of users currently logged into the platform, the number of active eNodeBs, and also the total number of users with access to the application. **Figure 20** presents the traffic visualization interface per eNodeB or per cell. It can be seen the evolution of traffic data generated over time for each eNodeB in **Figure 21** and **Figure 22** for a selected eNodeB.

In **Figure 22**, on the left, It can be seen the evolution of voice traffic for the sites of AHALA, Biyem-Assi and Nkomo (some sites located in Yaoundé town); and on the right, we have also the number of users simultaneously connected to each eNodeB.

All this information would help the operator to make better decisions regarding the optimization of his network, also to see the performance of each eNodeB (in terms of data consumption and resources) and to be able to identify the most profitable. Then RRC value can be adjusted based on users' demands. **Figure 23** presents the prediction visualization interface.

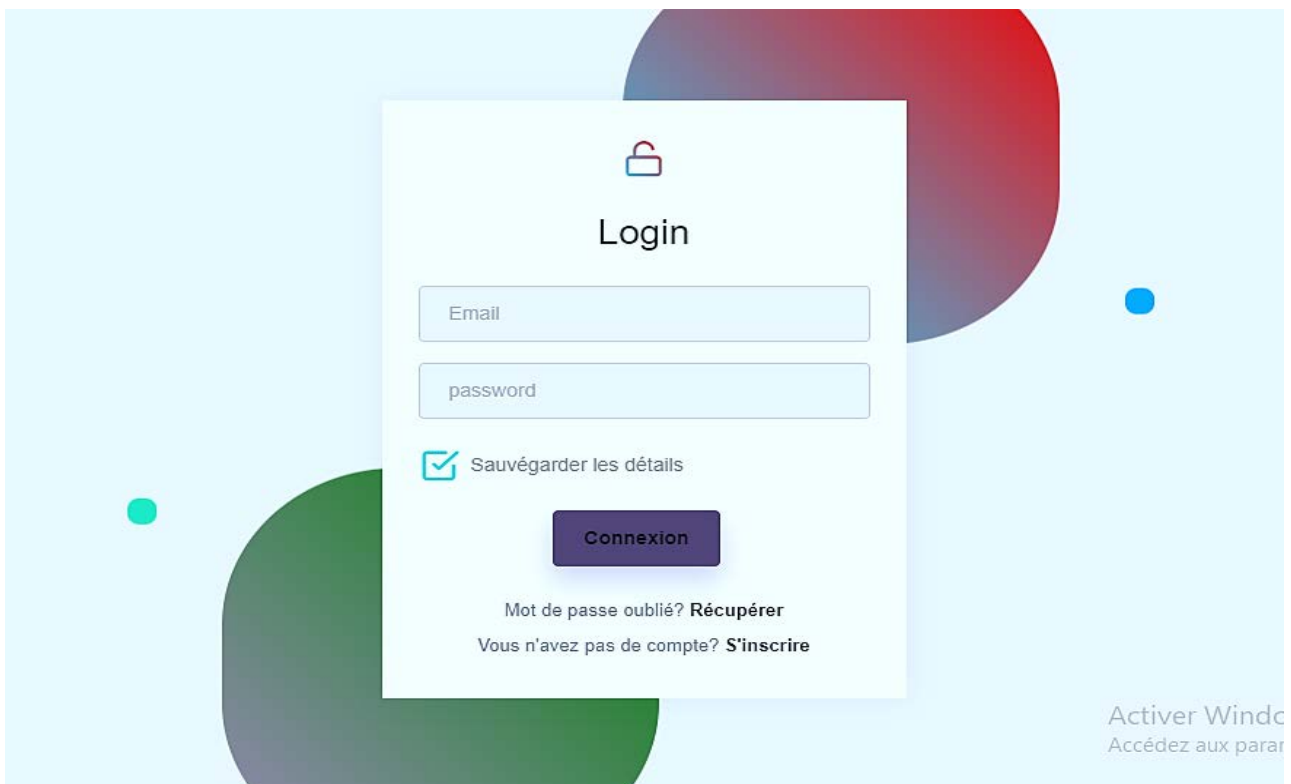


Figure 19. Login interface.

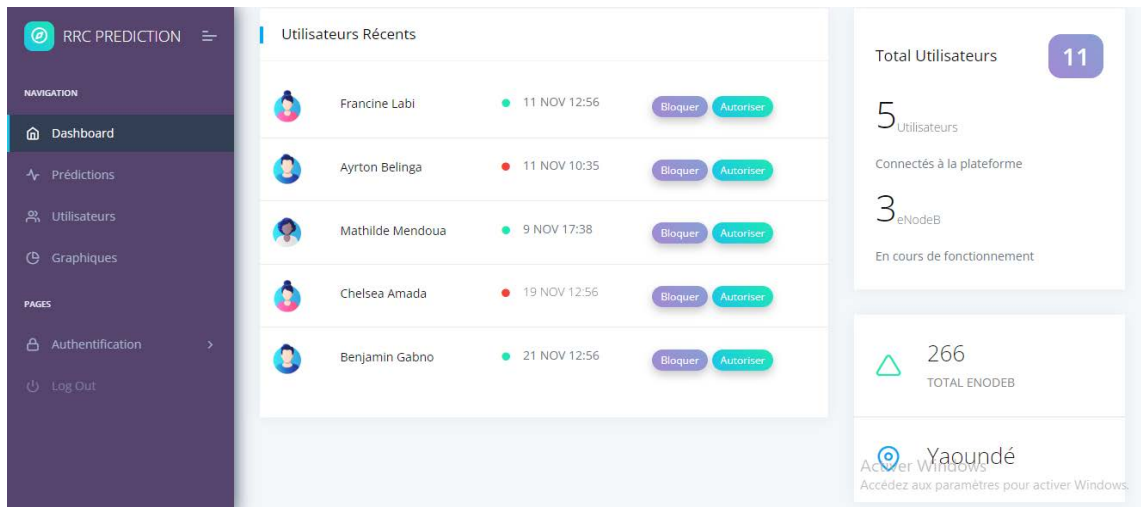


Figure 20. Dashboard presentation.

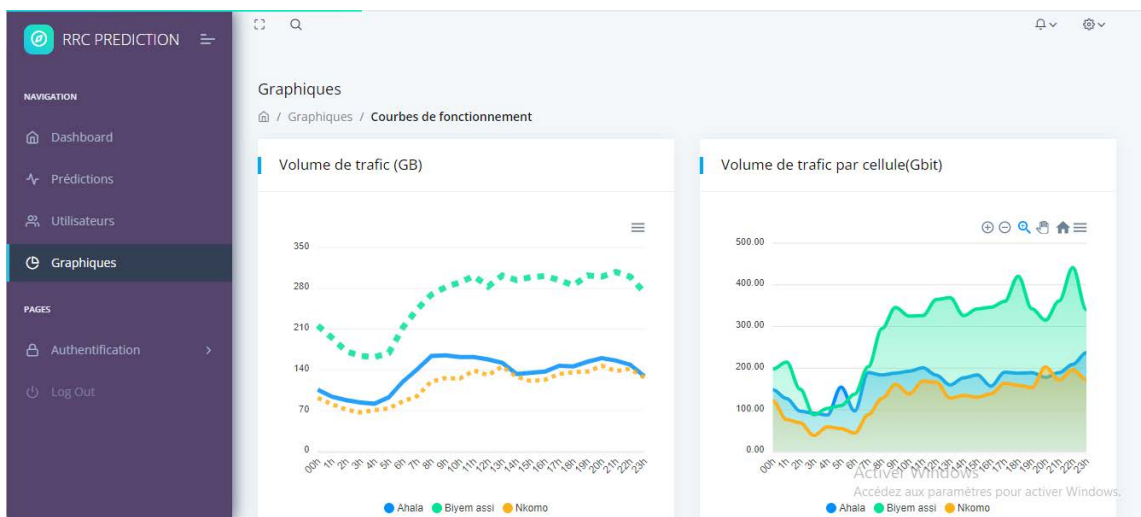


Figure 21. Traffic visualization interface 01.

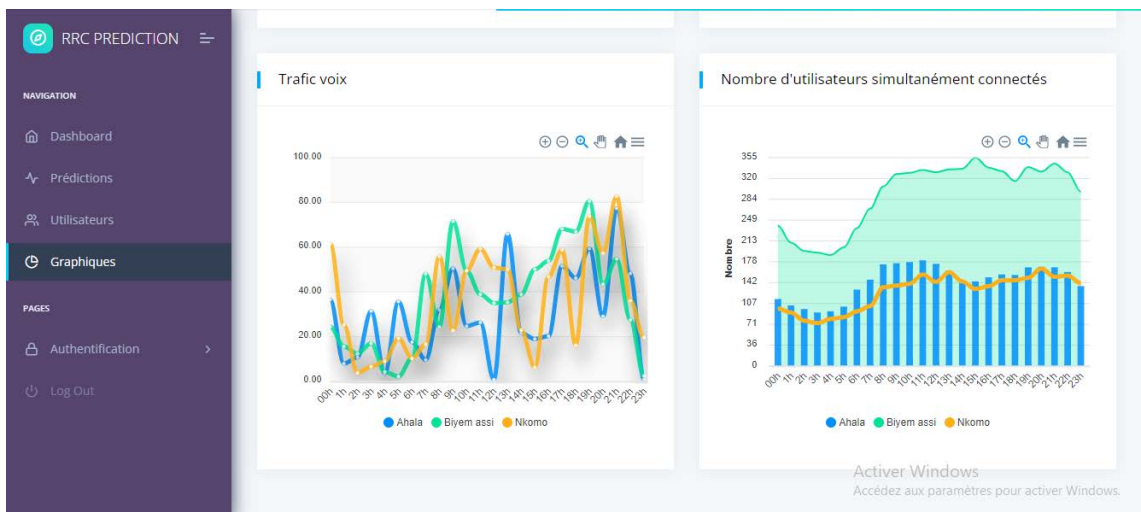


Figure 22. Traffic visualization interface 02.

NAME OF eNode B	TIME	REAL VALUES	PREDICT VALUES
eNodeB AHALA	00:00 02/07/2022	174.0	173.56
eNodeB NKOMO	00:00 02/07/2022	20.0	19.93
eNodeB BIYEM-ASSI	00:00 02/07/2022	330.0	329.97

Figure 23. Prediction interface.

To make a new prediction by using the proposed tool, simply import a file containing all the data consumption fields and the field to be predicted from the application; the trained model will directly recognize the fields and return the results of the predictions as soon as you click on the predict button as visible on the table.

5. Conclusions

The present research work focuses on dynamic resource allocation in LTE using machine learning which aims to solve the problem of resource underutilization due to static resource allocation at the eNodeB level. The objective set was to implement an efficient model that takes traffic data as input and accurately predicts the number of RRC to be allocated to each eNodeB. In this way, instead of having RRC resources per BTS like is done now in existing networks, the MNO could purchase RRC resources in bloc, and the license for dynamic allocation (This can be added in SON function of the OMC), then open the RRC dynamic allocation function in his network to enable the system to operate and assign RRC to eNodeBs automatically without any human intervention.

In order to reach that goal, machine learning algorithms were studied for resource allocation and implemented in Python by taking into account a certain number of features (volume of consumed data, user number, number of PRBs, etc.) which allow to make predictions, and ensure a precise allocation of resources. Finally, a web application was developed for the visualization of the various results. The results obtained from the development of the proposed model revealed that the model trained with the Long Term Short Term Memories (LSTM) is the most accurate among the three we proposed, with an accuracy of 97% against 47% for the linear model and 53% for the convolutional model. The latter performed better because it was better adapted to our data and con-

sequently deployed in our RRC Prediction application. To put our research into practice, Python was used to design the proposed model, Django for the interactions in the proposed application and HTML; CSS and JavaScript were used to develop the application's interfaces. We encountered difficulties in choosing the traffic-related fields for predicting the number of resources. In order to improve the efficiency of the proposed solution, we plan to adjust the model to support seasonal effects (Christmas, New Year's Day, Labor Day, National Day, etc.). The solution proposed in this paper can be used to expand Self Organizing Network functionalities and options in 4G and even 5G networks by vendors like Nokia, Huawei, ZTE and Ericson for the benefit of mobile network operators worldwide.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Ghosh, J. and Roy, S.D. (2017) The Implications of Cognitive Femtocell Based Spectrum Allocation over Macrocell Networks. *Wireless Personal Communications*, **92**, 1125-1143. <https://doi.org/10.1007/s11277-016-3597-x>
- [2] Deussom Djomadji, E.M., Fouba, B.A.R. and Kenfack Wamba, J.G. (2022) Performance Evaluation of the eICIC Technique Applied to a Heterogeneous 4G Mobile Network. *European Journal of Applied Sciences*, **10**, 540-560. <https://doi.org/10.14738/aivp.102.12244>
- [3] <https://www.mordorintelligence.com/fr/industry-reports/long-term-evolution-lte-market-growth>
- [4] Bahreyni, M.S. and Sattari-Naeini, V. (2014) Fairness Aware Downlink Scheduling Algorithm for LTE Networks. *Journal of Mathematics and Computer Science*, **11**, 53-63. <https://doi.org/10.22436/jmcs.011.01.06>
- [5] Habaebi, M.H., Chebil, J., Al-Sakkaf, A.G. and Dahawi, T.H. (2013) Comparison between Scheduling Techniques in Long Term Evolution. *IJUM Engineering Journal*, **14**, 67-76. <https://doi.org/10.31436/iiumej.v14i1.354>
- [6] Lu, Z., Yang, Y., Wen, X., Ju, Y. and Zheng, W. (2010) A Cross-Layer Resource Allocation Scheme for ICIC in LTE-Advanced. *Journal of Network and Computer Applications*, **34**, 1861-1868. <https://doi.org/10.1016/j.jnca.2010.12.019>
- [7] Tamwa, J.M., Tonye, E., Binele Abana, A. and Mveh, C. (2022) Intrusion Detection aided by Artificial Intelligence (IDAI). *American Journal of Engineering Research (AJER)*, **11**, 99-106.
- [8] Deussom, E., Matemtsap Mbou, B., Tchagna Kouanou, A., Ekonde Sone, M. and Bayonbog, P. (2022) Machine Learning-Based Approach for Designing and Implementing a Collaborative Fraud Detection Model through CDR and Traffic Analysis. *Transactions on Machine Learning and Artificial Intelligence*, **10**, 46-58.
- [9] Deussom Djomadji, E.M., Kabiena, I.B., Tchappa Tchito, C., Kouam Djoko, F.V. and Sone, M.E. (2023) Machine Learning-Based Approach for Identification of SIM Box Bypass Fraud in a Telecom Network Based on CDR Analysis: Case of a Fixed and Mobile Operator in Cameroon. *Journal of Computer and Communications*, **11**, 142-157. <https://doi.org/10.4236/jcc.2023.112010>

-
- [10] Batchakui, B., Deussom Djomadji, E.M., Chana, A. and Mama Tsimi, S.F. (2022) Comparing Machine Learning Algorithms for Improving the Maintenance of LTE Networks Based on Alarms Analysis. *Journal of Computer and Communications*, **10**, 125-137. <https://doi.org/10.4236/jcc.2022.1012010>
- [11] Deussom Djomadji, E.M., Takembo, T.C., Tchappa Tchito, C., Mamadou, A. and Sone, M.E. (2023) Machine Learning-Based Alarms Classification and Correlation in an SDH/WDM Optical Network to Improve Network Maintenance. *Journal of Computer and Communications*, **11**, 122-141. <https://doi.org/10.4236/jcc.2023.112009>
- [12] Zhou, Y., et al. (2015) Using Bidirectional Lstm Recurrent Neural Networks to Learn High-Level Abstractions of Sequential Features for Automated Scoring of Non-Native Spontaneous Speech. 2015 *IEEE Workshop on Automatic Speech Recognition and Understanding*, Scottsdale, AZ, USA, 13-17 December 2015, 38-345.
- [13] Hartmann, F.G., Kopp, J. and Lois, D. (2023) Linear Regression. In: Hartmann, F.G., Kopp, J. and Lois, D., Eds., *Social Science Data Analysis: An Introduction*, Springer, Wiesbaden, 87-112. <https://doi.org/10.1007/978-3-658-41230-2>
- [14] Crowley, J.L. (2023) Convolutional Neural Networks. In: Chetouani, M., Dignum, V., Lukowicz, P. and Sierra, C., Eds., *Human-Centered Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 13500, Springer, Cham, 67-80.
- [15] Čepukaitytė, G., Thom, J.L., Kallmayer, M., Nobre, A.C. and Zokaei, N. (2023) The Relationship between Short- and Long-Term Memory Is Preserved across the Age Range. *Brain Sciences*, **13**, Article No. 106. <https://doi.org/10.3390/brainsci13010106>