Deep Learning Based Load Balancing for improved QoS towards 6G

Vishnu Vardhan Nimmalapudi¹, Ajith Kumar Mengani¹, Roopa Vuppula², Rahul Jashvantbhai Pandya³ ¹,²,³ Department of Electronics and Communication Engineering, NIT Warangal, India

Email: ¹nvishnu@student.nitw.ac.in, ¹majith@student.nitw.ac.in, ²roopav44@student.nitw.ac.in, ³rpandya@nitw.ac.in,

Abstract

Deep learning has made great strides lately with the availability of powerful computing machines and the advent of user-friendly programming environments. It is anticipated that the deep learning algorithms will entirely provision the majority of operations in 6G. One such environment where deep learning can be the right solution is load balancing in future 6G intelligent wireless networks. Load balancing presents an efficient, cost-effective method to improve the data process capability, throughput, and expand the bandwidth, thus enhancing the adaptability and availability of networks. Hence a load balancing algorithm based on Long Short Term Memory (LSTM) deep neural network is proposed through which the base stations coverage area changes according to geographic traffic distribution, catering the requirement for future generation 6G heterogeneous network. The LSTM models performance is evaluated by considering three different scenarios, and the results were presented. Load variance coefficient (LVC) and load factor (LF) are introduced and validated over two wireless network layouts (WNL) to study the Quality of Service (QoS) and load distribution. The proposed method shows a decrease of LVC by 98.311% and 99.21% for WNL1, WNL2 respectively.

Index Terms

6G, Deep Learning, Long Short Term Memory, Load Balancing, Quality of Service.
I. INTRODUCTION

The fifth-generation (5G) of wireless technology was commercially deployed across 34 countries as of Jan. 2020. Hence research has begun to examine beyond 5G and to gestate the sixth-generation (6G). Deep Learning algorithms had shown promising results in the past few years in the field of communication systems [1], [2], motivating future 6G wireless networks to make use of these algorithms. The artificial intelligence (AI) enabled 6G intelligent network [3] is expected to be deployed between 2027 and 2030. The 6G wireless system needs to be introduced with new captivating features by using new technologies and simultaneously continuing the trends of the previous generations. The most powerful technologies and services like unmanned aerial vehicles (UAV), AI, 3D networking, autonomous vehicles, smart wearables, terahertz (THz) band, implants, sensing, 3D mapping, optical wireless communication (OWC), wireless power transfer, and computing reality devices, will be the driving force for 6G [4]. With the rapid growth of various emerging applications, such as virtual reality, Internet of Everything (IoE), and the three-dimensional (3D) media that require high data rates, the design goals for 6G, its implementation strategies, and challenges are already being explored in literature [5]–[8]. 5G can reach up to 20 Gb/s per-user bit rate for end-users with millimeter-wave (mm-wave) communications and large-scale antenna arrays, i.e., massive multiple-input and multiple-output (MIMO). Meanwhile, it is envisioned that the terahertz (THz) band in 6G will serve as the next frontier for communications with a per-user bit rate of approximately 1Tb/s in most cases [9]. Hence, the capacity to handle huge volumes of data and provide high-data-rate connectivity per device is the essential requirement for 6G intelligent networks. The load balancing technique can serve as a promising solution to efficiently handle higher data rates and manage wireless resource allocation among multiple connections. It also offers improved system performance, higher resource utilization, and decreased operational cost.

There are several methods in the literature for load balancing. S. T. Girma et al. proposed a fuzzy logic-based load balancing algorithm [10]. In this system, the traffic load is balanced by transferring some of the ongoing calls of densely loaded Base Transceiver Stations (BTS) to the underloaded BTS. The handoff index is calculated for the serving BTS and all its neighboring BTS, and once the neighboring BTS is identified better, the handoff process is executed. The
minimum value of this fuzzy system’s output is 0, which means no handoff, and the maximum value is 1, indicating exactly handoff. Y. Bejerano presented a load balancing scheme based on controlling the size of Wireless Local Area Network (WLAN) cells as like cell breathing [11]. It uses two algorithms: first is to decrease the load of congested AP(s), and the second is to provide an optimal min-max load-balanced solution. While J. Wu compare coverage shaping and Adaptive tilting based load balancing techniques [12]. It is proved that the coverage shaping based approach gives the best performance enhancement compared to the adaptive tilting system since the Bubble Oscillation Algorithm (BOA) [13] in coverage shaping technique finds the optimal boundaries to serve the massive demand. R. Misra et al. presented the machine learning oriented dynamic cost factors based routing in communication networks [14].

The above-mentioned load balancing techniques did not consider traffic priority as a criterion for balancing the load. Instead, they have only considered the traffic density. They are based on balancing the load by changing the coverage area for every few seconds, which may result in some power loss. Hence there is a trade-off between the power loss due to frequent change in coverage areas and the efficiency of load balancing. Therefore a predetermined load balancing technique is required through which the coverage area can be changed only a few times without changing the load balancing efficiency by much. Hence we propose a load balancing technique based on prioritized traffic prediction through which the area that should be covered by the base stations is predicted one day ahead. The load among the base stations is divided almost equally. The network traffic is prioritized into three default Quality of Service (QoS) Classes: High Priority (Priority-1 or P1), Medium Priority (Priority-2 or P2), and Low Priority (Priority-3 or P3). The traffic, which is essential or time-critical and needs to be processed without any delay, is considered a high priority. Generally, military communications, banking transactions, and videos on Amazon Prime, Netflix, YouTube, video games live-streaming sites are considered as high priority, whereas emails, Facebook, Whatsapp are considered as Medium Priority, and unessential browsing data is considered as Low Priority.

There are several methods available in the literature for wireless traffic modeling and prediction. Y. Shu et al. applied the most widely used time series forecasting method, Auto-Regressive Integrated Moving Average (ARIMA), for traffic load prediction [15]. S. Jaffry proved that the Long short-term memory (LSTM) and vanilla feed-forward neural networks (FFNN) could
predict the cellular data traffic more accurately than the statistical ARIMA model [16]. While C. Zhang et al. predicted the citywide traffic using Convolutional Neural Network (CNN) by treating data traffic as images [17]. However, the approaches mentioned above for traffic prediction did not explain the performance of their system for the scenario where there is a sudden change in traffic pattern compared to daily patterns due to situations like pandemics or summer holidays where the traffic pattern would be entirely different in many areas compared to regular days. In this paper, we propose a model for traffic prediction that can adapt to any situation in less time. The layout of the remaining paper is as follows. Section II delineates the System’s Model. Section III contains an analysis of our experiments and results. Section IV illustrates the conclusions and discusses future areas of exploration.

II. SYSTEM MODEL

Our system model mainly comprises of two parts: Cellular Traffic Predictor (CTP) and Premeditated Cell Transformer (PCT). The main aim of CTP is to predict the traffic and the priorities that would be generated the next day from a given area when trained with the previous traffic and priority data. While PCT trades the coverage area between every neighboring Base Station (BS) pair until the load is divided equally among all the Bases stations, thereby predicting the area that should be covered by each BS the next day. Generally, while designing a cellular network, the geographical area is divided into small hexagonal regions called cells, and at the center of hexagons, BSs are installed. Hexagonal cells are preferred instead of circular cells since in hexagonal cells, frequency reuse is feasible, and they can cover the entire area without overlapping. Each of these cells is divided into 24 microcells in a triangular shape, as shown in Fig.1. The circles in the figure depict the maximum area that can be covered by the BS’s. Massive MIMO technology is used since these antenna groups can have the directivity up to 360° with better throughput and spectrum efficiency. The microcells which are inside the circle and are exterior to the hexagon area of a BS are called the Scope-1 microcells with respect to the corresponding BS. Whereas Scope-0 microcells w.r.t a BS are those microcells which are inside the hexagonal area covered by that BS and are at the border to the neighboring hexagons. Hence after predicting the traffic of each of these microcells, to balance the load, neighboring BSs exchange some of their Scope-1 and Scope-0 microcells until both carry the equal load.
A. Architecture and Hypothesis

The architecture of the system model is shown in Fig.2. Consider data vectors $T_t, T_{t-1}, ..., T_{t-p}$ containing the time-series data traffic of each microcell at timesteps $t, t-1, ..., t-p$ as given in Eq.(1-3)

$$T_t = \{T_{1,t}, T_{2,t}, ..., T_{n,t}\}$$
\[ T_{t-1} = \{T_{1,t-1}, T_{2,t-1}, \ldots, T_{n,t-1}\} \]  
(2)

\[ T_{t-p} = \{T_{1,t-p}, T_{2,t-p}, \ldots, T_{n,t-p}\} \]  
(3)

where \((p+1), n\) represents the sequence length and the total number of microcells respectively. Total number of BSs \(N = \frac{n}{24}\). The data traffic of every microcell across all timesteps is represented as \(T\). In general, \(T_{i,j}\) represents the traffic value at \(i^{th}\) microcell at \(j^{th}\) timestep. Similarly, \(P_{1,i,j}, P_{2,i,j}, P_{3,i,j}\) represents the percentage of traffic at Priority-1, Priority-2, Priority-3 respectively at \(i^{th}\) microcell at \(j^{th}\) timestep, and the traffic, priorities data across all timestamps are represented as \(T, P_1, P_2, P_3\) and are given in Eq.(4-7).

\[ T = \{T_t, T_{t-1}, \ldots, T_{t-p}\} \]  
(4)

\[ P_1 = \{P_{1,t}, P_{1,t-1}, \ldots, P_{1,t-p}\} \]  
(5)

\[ P_2 = \{P_{2,t}, P_{2,t-1}, \ldots, P_{2,t-p}\} \]  
(6)

\[ P_3 = \{P_{3,t}, P_{3,t-1}, \ldots, P_{3,t-p}\} \]  
(7)

Since a given microcell at any given timestep, the sum of all the priorities is equal to 100%, Eq.(8) should be satisfied.

\[ P_{1,i,j} + P_{2,i,j} + P_{3,i,j} = 1 \]  
(8)

The CTP predicts the next timestep traffic and priority data of each microcell by taking in their previous data and is presented in Eq.(9-12).

\[ T_{t+1} = CTP_1(\{T_t, T_{t-1}, \ldots, T_{t-p}\}) \]  
(9)

\[ P_{1,t+1} = CTP_2(\{P_{1,t}, P_{1,t-1}, \ldots, P_{1,t-p}\}) \]  
(10)

\[ P_{2,t+1} = CTP_3(\{P_{2,t}, P_{2,t-1}, \ldots, P_{2,t-p}\}) \]  
(11)

\[ P_{3,t+1} = CTP_4(\{P_{3,t}, P_{3,t-1}, \ldots, P_{3,t-p}\}) \]  
(12)

where \(CTP_1, CTP_2, CTP_3, CTP_4\) are the functions that fit the LSTM deep neural networks. Since the predicted priorities do not always satisfy Eq.(8), it can be presumed that they satisfy Eq.(13),

\[ P_{1,i,t+1} + P_{2,i,t+1} + P_{3,i,t+1} \neq 1 \]  
(13)
To make them satisfy Eq.(8), they are activated using softmax function and the corresponding activated outputs $P'_{1,t+1}$, $P'_{2,t+1}$, $P'_{3,t+1}$ presented in Eq.(14-16),

$$P'_{1,t+1} = \frac{\exp P_{1,i,t+1}}{\exp P_{1,i,t+1} + \exp P_{2,i,t+1} + \exp P_{3,i,t+1}}$$ (14)

$$P'_{2,t+1} = \frac{\exp P_{2,i,t+1}}{\exp P_{1,i,t+1} + \exp P_{2,i,t+1} + \exp P_{3,i,t+1}}$$ (15)

$$P'_{3,t+1} = \frac{\exp P_{3,i,t+1}}{\exp P_{1,i,t+1} + \exp P_{2,i,t+1} + \exp P_{3,i,t+1}}$$ (16)

It is clear from Eq.(3, 4, and 5) that, activated outputs satisfy Eq.(17).

$$P'_{1,t+1} + P'_{2,t+1} + P'_{3,t+1} = 1$$ (17)

Generally, the uplink bandwidth allotted for $P1$ is four times the bandwidth allotted for $P3$ and two times the bandwidth allotted for $P2$ [18]. Hence it is considered that $P1$ is four times more important than $P3$ and two times more important than $P2$ for the following analysis. Since the analysis of distributing the load among all BSs by considering traffic, $P1$, $P2$, $P3$ data of each microcell is severe; a new parameter called traffic score($S$) is introduced that depends on each of the traffic, $P1$, $P2$, $P3$ data. The traffic score for for $i^{th}$ cell and for all cells are given in Eq.(18),(19) respectively.

$$S_{i,t+1} = 4 * T_{i,t+1} * P'_{1,i,t+1} + 2 * T_{i,t+1} * P'_{2,i,t+1} + T_{i,t+1} * P'_{3,i,t+1}$$ (18)

$$S_{t+1} = \{S_{1,t+1}, S_{2,t+1}, ..., S_{n,t+1}\}$$ (19)

Based on the traffic scores($S_{t+1}$) and the location data of microcells, load balancing is done by PCT.

B. Cellular Traffic Predictor

The CTP consists of a multistage LSTM deep neural network capable of predicting data traffic and priorities data based on previous data. LSTMs [19] are a special kind of Recurrent Neural Network (RNN), capable of remembering information for long periods. Each LSTM layer has a chain-like structure with the repeating modules called LSTM-cells. The structure of LSTM-cell
is shown in Fig. 3. Here, $I_t$ is the input to the current cell, $y_t, y_{t-1}$ denotes current and previous hidden states and $C_t, C_{t-1}$ denotes the current and previous cell states. There are three different gates in an LSTM-cell that control the flow of information. They are: forget gate ($\Gamma_t$), input gate ($i_t$), and output gate ($\Omega_t$). The forget gate removes the information, which is less important and keeps only the information that is required for understanding things. The input gate is responsible for adding information to the cell state while the output gate decides what the next hidden state should be.

![Fig. 3: LSTM cell.](image)

Source: [20]

The expressions for all the gates, hidden state, cell state are as given in Eq.(20-25),

\[
\Gamma_t = \text{Sig}(U_f[y_{t-1}, I_t] + \phi_f) \\
i_t = \text{Sig}(U_i[y_{t-1}, I_t] + \phi_i) \\
a_t = \text{tanh}(U_a[y_{t-1}, I_t] + \phi_a) \\
C_t = \Gamma_t \ast C_{t-1} + i_t \ast a_t \\
\Omega_t = \text{Sig}(U_o[y_{t-1}, I_t] + \phi_o) \\
y_t = \Omega_t \ast \text{tanh}(C_t)
\]
Fig. 4: Deep Neural Network architecture used in CTP.

Where $\text{Sig}$ represents the sigmoid function which is given by $\text{Sig}(x) = \frac{1}{1+\exp(-x)}$ and $\ast$ denotes the element-wise multiplication. While $U(.)$ and $\phi(.)$ denote the vectors of weights and biases corresponding to the respective gates, input, hidden layer. The architecture of the LSTM deep neural network used for this work is shown in Fig.4.

C. Premeditated Cell Transformer

After predicting the traffic score values of all the microcells using the LSTM network, the total score is distributed equally among all the BSs using PCT. The basic principle involved in PCT is, the exchange of Scope-1 and Scope-0 microcells between adjacent hexagons until all the
Algorithm 1: Premeditated Cell Transformer

\textbf{S}: Set of traffic scores of all triangles where \(S[i]\) represents the traffic score of \(\text{quotient}(i \div 24)\)\textsuperscript{th} hexagon’s \(\text{remainder}(i \div 24)\)\textsuperscript{th} triangle;

\(N\): Number of Hexagons/BSs;

\(HL\): Hexagon’s location data;

\(\text{avg\_score} = 0\);

\(\text{iterations} = 0\);

\textbf{Function} \texttt{GetNearByHexagons}(num):

\textbf{for} \(i \leftarrow 1 \text{ to } N \) \textbf{do}

\textbf{if} euclidian distance \((HL[i][0], HL[num][0]) = 4 \times \text{height of triangle and } i \neq \text{num} \) then

\hspace{1cm} Add to hexagon;

\textbf{return} hexagon;

\textbf{Function} \texttt{rec}(addable, min\_index, u, added, HexTraffic):

\hspace{1cm} i, j = addable(u);

\hspace{1cm} if \(u + 1 < \text{length of addable} \) then

\hspace{2cm} return HexTraffic, Added;

\hspace{1cm} if \(\text{HexTraffic}[\text{min\_index}] + S[(i-1) \times 24 + j] > \text{avg\_score} \text{ or } \text{HexTraffic}[i] + S[(i-1) \times 24 + j] < \text{avg\_score} \) then

\hspace{2cm} return \texttt{rec}(addable, min\_index, u + 1, added, HexTraffic);

\hspace{1cm} HexTraffic1, Added1 = \texttt{rec}(addable, min\_index, u + 1, added, HexTraffic);

\hspace{1cm} HexTraffic2, Added2 = \texttt{rec}(addable, min\_index, u + 1, added + (i, j), HexTraffic);

\hspace{1cm} if \(\text{HexTraffic1}[\text{min\_index}] < \text{HexTraffic2}[\text{min\_index}] \) then

\hspace{2cm} return HexTraffic2, Added2;

\hspace{1cm} return HexTraffic1, Added1;

\textbf{Function} \texttt{GetAddableTriangles}(num):

\textbf{addable} = [];

\textbf{nearby} = \texttt{GetNearByHexagons}(num);

\textbf{for each} \(i \in \text{nearby} \) \textbf{do}

\hspace{1cm} if \(j\text{th triangle of }i\text{th hexagon falls under scope 1 of }num\) then

\hspace{2cm} add \((i, j)\) to the addable list;

\textbf{return} addable;

\textbf{Function} \texttt{TrafficSharingAlgo}(S, N, HL):

\hspace{1cm} HexTraffic, Added be two new arrays.

\textbf{for} \(i \leftarrow 1 \text{ to } N \) \textbf{do}

\hspace{1cm} \textbf{HexTraffic}[i] = \text{sum of values from } S[(i-1) \times 24 + 1] \text{ to } S[(i-1) \times 24 + 24];

\hspace{1cm} n = \text{length of } S;

\hspace{1cm} \text{avg\_score} = \Sigma_{i=1}^{n} S[i] / N;

\hspace{1cm} L = [1, 2, \ldots, N] \text{ be a new array mapped to array HexTraffic such that } L[k] \text{ maps to } \text{HexTraffic}[k];

\hspace{1cm} \text{sort elements of } L \text{ in increasing order of their corresponding mappings in HexTraffic} ;

\textbf{for each} \(j \in L \) \textbf{do}

\hspace{1cm} addable = \texttt{GetAddableTriangles}(j);

\hspace{1cm} remove the elements from addable that are present in Added array;

\hspace{1cm} \text{HexTraffic, NewAdded} = \texttt{rec}(addable, j, 0, Added, HexTraffic);

\hspace{1cm} Added = Added + NewAdded;

\hspace{1cm} \text{iterations} = \text{iterations} + 1;
hexagons carry nearly equal traffic scores. In order to distribute the total traffic score effectively, the following steps are used,

**Step 1.** Find the traffic score under each hexagon (BS).

**Step 2.** Compute the average traffic score based on number of hexagons \( N \) which is presented in Eq.(26).

\[
\text{Average Traffic Score} = \frac{\sum_{i=1}^{n} S_{i,t} + 1}{N} \tag{26}
\]

**Step 3.** Store the hexagons which have traffic score less than the average score in a list.

**Step 4.** Take a hexagon from the obtained list in Step3, which has the least traffic and find out all the scope1 triangles for this hexagon. **Step 5.** Try all possible selections of triangles obtained from Step 4 and add the set of triangles with maximum possible traffic to the hexagon, such that the traffic score after adding does not exceed the average traffic score and traffic score of the base tower from which we take does not become less than average traffic score. Discard this hexagon from the list so that it does not involve in further steps. **Step 6.** Repeat Step 4 and Step 5 until the list obtained from Step 3 becomes empty. The Algorithm 1 describes the implementation of Premeditated Cell Transformer. Here all the hexagons are labelled with numbers starting from...
Each hexagon contains 24 triangles numbered from 1 to 24. The traffic of \( j^{th} \) triangle of \( i^{th} \) hexagon is given by \( S[i \times 24 + j] \). All the location co-ordinates of hexagon are stored in \( HL \), where \( HL[i] \) represents all co-ordinates corresponding to \( ith \) hexagon. \( HL[i][0] \) contains the \( i^{th} \) hexagon center location co-ordinates and \( HL[i][j] \) contains \( j^{th} \) triangle’s location co-ordinates of \( i^{th} \) hexagon. The function \( rec \) in the algorithm 1, does backtracking to find out the best possible selection of triangles, updates the traffic score values and assigns areas(triangles) to the selected low traffic score tower. The term \( iterations \) in the algorithm 1 indicates the number of hexagons that have reached saturation and will not involve in traffic sharing in the further steps.

The analysis of PCT is performed by considering two wireless network layout (WNL) examples. Fig.5 consists of three BSs and all their triangles. Each triangle area’s traffic score is also shown. The triangles having the same color get served by the same base tower. The image shows that, after applying the algorithm, some of the traffic (Triangle area) from base towers having
higher traffic scores, are taken by base towers with lower traffic scores. Before optimization, the approximate traffic scores under the base towers are 1529.24, 1356.02, 1661.87, and after optimization, the values are 1529.24, 1502.008, 1515.8195 thereby decreasing the variance of
all the traffic scores making the network optimized. Fig. 6 consists of seven base towers and all their triangles. Each triangle area’s traffic score is also shown. The triangles having the same color get served by the same base tower. The image shows us that after applying the algorithm, few of the traffic (Triangle area) from base towers having a higher traffic score is taken by base towers having lower traffic scores. Before optimization, the approximate traffic scores under the base towers are 1571.1097, 1266.1715, 1199.427, 1003.5461, 1438.0893, 1316.9314, 1309.004 and after optimization, the values are 1301.5125, 1266.1715, 1294.6283, 1288.0092, 1342.8885, 1302.0655, 1309.0043 thereby decreasing the variance of all the traffic scores making the network optimized.

III. EXPERIMENTS AND RESULTS

Since there is no dataset available online containing all traffic, P1, P2, P3 data in one place, we prepared our dataset. The proposed architecture is trained and tested through a random dataset generated using a Gaussian distribution [21]. For each triangle, after considering some random base values, for each of the traffic, P1, P2, P3, random values generated using Gaussian
distribution with mean as 0 and standard deviation as one are added to this base value generate the whole dataset. The probability density for a gaussian distribution is given in Eq.(27).

\[
p(x) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)
\]

(27)

Where \( \sigma, \mu \) represents the standard deviation and the mean, respectively. Evaluating the performance of the model is tested with three scenarios to check how well it can adapt to situations. In scenario-1, it is considered that the data traffic does not show much variation between weekdays and weekends. While in scenario-2, the data traffic shows variation between weekdays and weekends. In scenario-3, the traffic may change due to sudden changes like vacations, pandemic situations, etc. Scenario-3 is again classified into two classes, where in the first, the sudden change of traffic happens in the distant past, while in the second, the sudden change of traffic happens in the recent past.

Furthermore, the dataset contains 2800 training samples for each of the traffic, P1, P2, P3. It comprises 700 training samples from each scenario-1 and scenario-2 and the remaining 1400 from scenario-3. While each test set contains 77 samples, each representing data traffic for each day, of which the model predicts the traffic from the 50th sample to the 77th sample.
prediction results for data traffic are shown in Fig.7. Scenario-2 is explained with two cases where in the first case, the data traffic is high on weekends compared to weekdays, and in the second case, the data traffic is low on weekends compared to weekdays. Similarly, for scenario-3, in the first case, data traffic decreases due to sudden changes, while in second, the data traffic increases due to sudden changes. It can be observed from Fig.7(c) that the model can adapt to the new situations very well within two to three days. The same model is used to predict the P1, P2, P3 data also.

Furthermore, Fig.8 and 9 show the traffic scores handled by each BS before and after applying PCT, considering the WNL shown in Fig.5 and Fig.6, respectively. To study the QoS and smoothness of the load distribution in the network, we define two parameters: The load variance
coefficient (LVC) and load factor (LF) and they are presented in Eq.(28),(29) respectively. LVC indicates the deviation in all the BSs’ traffic scores from the average traffic score, whereas LF is the ratio of the average traffic score to the peak traffic score.

\[
LVC = \frac{\sum_{j=1}^{N} \left( \sum_{i=24*(j-1)+1}^{24*(j-1)+24} S_{i,t+1} \right) - \sum_{i=1}^{N} S_{i,t+1}}{N-1}
\]

(28)

\[
LF = \frac{\text{AverageTrafficScore of a cell}}{\text{PeakTrafficScore of a cell}}
\]

(29)

The decrease in LVC and LF indicates that the traffic is distributed almost evenly among all BSs, thereby increasing the effective bandwidth and the QoS. The variation of LVC and LF for WNL1 and WNL2 are presented in Fig.10 and 11, which implies that there is a gradual decrease in LVC and LF as the iterations increases, thus improving the QoS. For WNL1, LVC decreases from 32013 to 540.2, and for WNL2, it decreases from 23524.6 to 185.5 while LF decreases from 1.2 to 1.03 for WNL1 and from 1.1 to 1.008 for WNL2.

IV. CONCLUSION

A novel load balancing scheme for 6G is proposed through which the traffic load is divided almost equally among all the BSs. It is based on two algorithms, CTP and PCT. CTP predicts the traffic, P1, P2, P3 data using the LSTM network while PCT distributes the total traffic score among all BSs. Distribution of load is done in such a way that the LVC shows a decrease of 98.31% and 99.21% for WNL1, WNL2, respectively. Results shown by LSTM and PCT infers that the performance of this method proposed is best or comparable to the existing methods. In the future, the load balancing analysis can be extended by considering the losses in the wireless channel that become significant with the distance.

REFERENCES

[1] R. V. Hemadri, A. Rayaluru, and R. J. Pandya, AEVB-Comm: An Intelligent Communication System based on AEVBs, arXiv.org, 19-May-2020. [Online]. Available: https://arxiv.org/abs/2005.09391. [Accessed: 27-Jun-2020].

[2] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, Deep Learning-Based End-to-End Wireless Communication Systems With Conditional GANs as Unknown Channels, IEEE Transactions on Wireless Communications, vol. 19, no. 5, pp. 31333143, 2020.

[3] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches, Proceedings of the IEEE, vol. 108, no. 2, pp. 292307, 2020.
[4] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, 6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions, arXiv.org, 25-Sep-2019. [Online]. Available: https://arxiv.org/abs/1909.11315. [Accessed: 27-Jun-2020].

[5] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, The Roadmap to 6G: AI Empowered Wireless Networks, IEEE Communications Magazine, vol. 57, no. 8, pp. 8490, 2019.

[6] Y. Zhao, G. Yu, and H. Xu, 6G mobile communication networks: vision, challenges, and key technologies, SCIENTIA SINICA Informationis, vol. 49, no. 8, pp. 963987, 2019.

[7] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond, IEEE Access, vol. 7, pp. 7872978757, 2019.

[8] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies: Semantic Scholar, undefined, 01-Jan-1970. [Online]. Available: https://www.semanticscholar.org/paper/6G-Wireless-Networks:-Vision,-Requirements,-and-Key-Zhang-Xiao/ffa6396d0749697fc11e1e24a5d9ce53f8bf10e1fc. [Accessed: 27-Jun-2020].

[9] S. Mumtaz, J. M. Jornet, J. Aulin, W. H. Gerstacker, X. Dong, and B. Ai, Terahertz Communication for Vehicular Networks, IEEE Transactions on Vehicular Technology, vol. 66, no. 7, pp. 56175625, 2017.

[10] S. T. Girma and A. G. Abebe, Mobility Load Balancing in Cellular System with Multicriteria Handoff Algorithm, Advances in Fuzzy Systems, vol. 2017, pp. 18, 2017.

[11] Y. Bejerano and S.-J. Han, Cell Breathing Techniques for Load Balancing in Wireless LANs, IEEE Transactions on Mobile Computing, vol. 8, no. 6, pp. 735749, 2009.

[12] J. Wu, J. Bigham, P. Jiang, and J. Neophytou, Tilting and Beam-shaping for Traffic Load Balancing in WCDMA Network, 2006 European Conference on Wireless Technologies, 2006.

[13] L. Du, J. Bioham, and L. Cuthbert, A bubble oscillation algorithm for distributed geographic load balancing in mobile networks, Ieee Infocom 2004.

[14] R. Misra and R. J. Pandya, Machine Learning oriented Dynamic Cost Factors based Routing in Communication Networks, Proc. International Conference on Communications, Signal Processing and VLSI (IC2SV2019), Warangal, India, 2019, pp.1-8.

[15] Y. Shu, M. Yu, J. Liu, and O. Yang, Wireless traffic modeling and prediction using seasonal ARIMA models, IEEE International Conference on Communications, 2003. ICC 03.

[16] S. Jaffry, Cellular Traffic Prediction with Recurrent Neural Network, arXiv.org, 05-Mar-2020. [Online]. Available: https://arxiv.org/abs/2003.02807. [Accessed: 27-Jun-2020].

[17] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, Citywide Cellular Traffic Prediction Based on Densely Connected Convolutional Neural Networks, IEEE Communications Letters, vol. 22, no. 8, pp. 16561659, 2018.

[18] Using Packet Prioritization on a Traffic Shaping Rule, Cisco Meraki, 01-Apr-2020. [Online]. Available: https://documentation.meraki.com/MX/Firewall_and_Traffic_Shaping/Using_Packet_Prioritization_on_a_Traffic_Shaping_Rule. [Accessed: 27-Jun-2020].

[19] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, Neural Computation, vol. 9, no. 8, pp. 17351780, 1997.

[20] K. Mani, GRU’s and LSTM’s, Medium, 18-Feb-2019. [Online]. Available: https://towardsdatascience.com/grus-and-lstm-s-741709a9b9b1. [Accessed: 27-Jun-2020].
[21] N. R. Goodman, Statistical Analysis Based on a Certain Multivariate Complex Gaussian Distribution (An Introduction), The Annals of Mathematical Statistics, vol. 34, no. 1, pp. 152177, 1963.