Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-025-00034-2

Original Article

# Enhancing Long-Term Evolution (LTE) Traffic Prediction Using Long Short-Term Memory Neural Networks

Temitope O. Efuwape [1, 2, ✉], Olusegun A. Ojesanmi [3], Temitope E. Ogunbiyi [4], Omolara O. Akinlade [5]

1. *Computer Science/AI/ML/Data Science & Analytics, Bells University of Technology, Ota, NGA*
2. *Computer Science/ML/Data Science & Analytics/Networks, Federal University of Agriculture, Abeokuta, NGA*
3. *Computer Science/Computer Systems and Networks, Federal University of Agriculture, Abeokuta, NGA*
4. *Computer Science, Software Engineering, and Cloud Computing, Bells University of Technology, Ota, NGA*
5. *University Library/Systems Librarian, Bells University of Technology, Ota, NGA*

Received: April 22, 2025 | Review began: May 04, 2025 | Review ended: March 13, 2026 | Published: March 18, 2026



## Abstract

Globally, the telecommunications sector has undergone significant transformations. Driven by rapid technological progress, increasing teledensity, and evolving consumer demographics, this industry is at a crucial juncture. These transformations are characterized by frequent deterioration in the quality of service (QoS) offered by telecommunications service providers, which results in poor experiences for consumers and inefficient network resources. To address this gap, this study investigates the application of deep learning techniques for long-term evolution (LTE) traffic demand prediction using real-world, anonymized cellular data obtained from a sub-Saharan African telecommunications service provider. Precisely, three long short-term memory (LSTM) neural network architectures, vanilla LSTM, stacked LSTM, and bidirectional LSTM, are developed and evaluated under different loss functions, with the Adam optimizer employed during training. The study demonstrates the significance of choosing an appropriate loss function based on the inherent characteristics of the problem, data distribution, and model constraints to enhance each prediction. The predictions were evaluated using the coefficient of determination $R^2$ metric. The mean squared error loss function was determined to be the most effective for LTE traffic demand prediction task, achieving an accuracy of 86.44% with the vanilla LSTM, while the bidirectional LSTM reached 85.43% accuracy by using the Smooth_L1 loss function during learning. The findings establish that increased architectural complexity does not necessarily translate into improved performance for LTE traffic demand prediction in data-constrained environments. Consequently, the vanilla LSTM is recommended as an effective and computationally efficient solution for traffic prediction and QoS-aware resource management in LTE networks within sub-Saharan Africa.

**Categories:** AI applications, Machine Learning (ML), Computer Networks

**Keywords:** lte, machine learning, long short-term memory, loss function, traffic prediction

## Introduction

Cellular networks have become a central component of industrialised society by offering dependable, higher-speed, and lesser-latency connectivity through a range of technologies, including Wireless Fidelity (WiFi) networks, 4G, and 5G, among others. The ever-going advancement and acceptance of these technologies have led to a notable increase in mobile users and data-intensive applications across various sectors such as education, media, banking, transportation, and manufacturing [1]. As reported by Ericsson [2], global mobile data traffic is projected to exceed 90 EB/month by 2022 and is continuously expected to grow to around 115 EB/month by the end of 2028. However, it presented considerable challenges for telecommunications operators in managing the extensive cellular traffic while at the same time ensuring and improving quality of service (QoS) and user experience. Additionally, the number of mobile data users is anticipated to grow significantly, with 7.3 billion wireless data devices in 2022 and a projected rise to over 9.2 billion by 2028 [2]. Consequently, there is a pressing need to address the challenges related to enhancing quality in LTE network services. The most simple solution is to install more eNodeBs to increase the capacity of cellular networks and achieve load balancing, but this approach incurs substantial costs for telecommunications operators.

In the realm of Long-Term Evolution (LTE) networks, machine learning (ML) has attracted considerable interest due to its capability to process and interpret vast amounts of data, identify significant patterns, and deliver precise predictions [3]. This process entails creating algorithms that can learn from data, adjust, and enhance their performance over time without the need for overt programming. Applications of ML techniques, including resource allocation, QoS, network traffic monitoring and analysis, anomaly detection, and cybersecurity, offer promising solutions for enhanced user experience and performance. One important subfield of network traffic monitoring and analysis in LTE networks is traffic prediction. Analysing historical data, current traffic trends, and various factors such as time of day, user behaviour, physical resource block utilisation data, and network real-time state, ML-based algorithms can predict future traffic demands with high accuracy. This enables network operators to judiciously allocate network resources and optimise performance and QoS provisioning. Figure *1* shows the generalised approach of the ML process. ML models for cellular traffic prediction can evaluate complex and large volumes of datasets and extract insightful information for informed decisions. In the management of LTE networks, deep learning neural networks have drawn a lot of attention. Neural networks in deep learning consist of many hidden layers of interconnected neurones, which are skilled at learning hierarchical data representation and abstraction such as images, sound, and text [4]. However, a variety of neural architectures have transformed the sphere of deep learning, with long short-term memory (LSTM) standing out for in-sequence data processing such as time series analysis. This study explores the importance of deep learning neural networks learning parameters like loss functions to enhance LSTM traffic demand prediction.

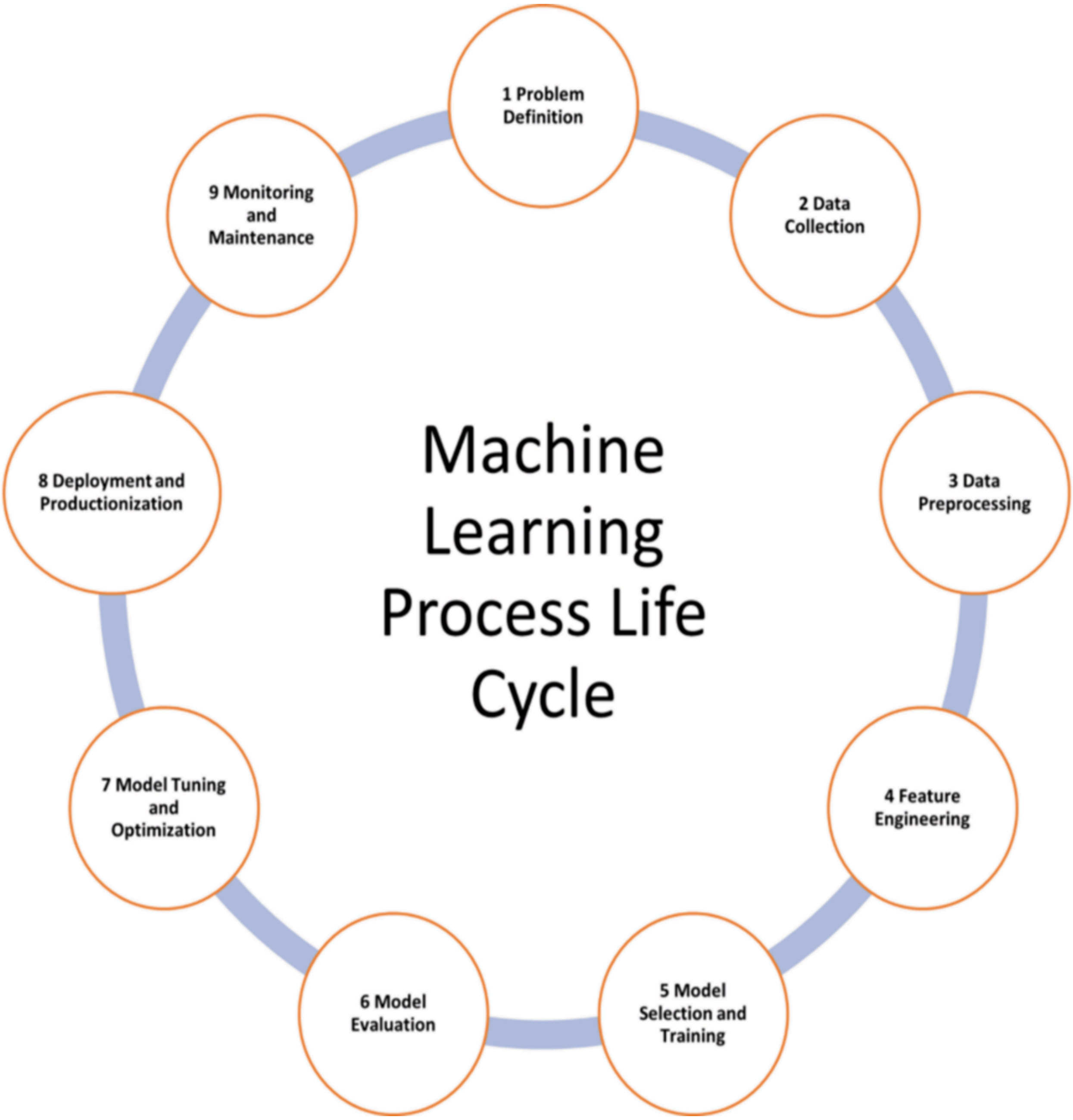


**FIGURE 1: Machine Learning Process Life Cycle**

## Related review

Network traffic prediction through ML has received significant growth in adoption because it creates valuable opportunities for enhancing QoS, detecting anomalies, load balancing, connection management, and optimising resource use [5-6]. Research indicates that various methods, including statistical techniques, traditional ML, and deep learning models, successfully contribute to managing LTE networks.

The research by Alsaade and Al-Adhaileh [7] applied statistical techniques for modelling and forecasting LTE network traffic volume. The testing method proved better than others did for traffic prediction purposes, though with lesser accuracy compared to ML techniques. Kuboye et al. [8] conducted a research study using different supervised learning models to predict traffic congestion in LTE networks through user perception data analysis. Researchers conducted an evaluation of k-Nearest Neighbour, Support Vector Machine, Decision Tree, and Logistic Regression. The analytical study confirmed k-nearest neighbour exceeds other ML approaches when it comes to LTE network traffic congestion prediction. The study determined that LTE network users mostly encounter traffic congestion at present. Bernabe et al. [9] conducted research on applying ML algorithms to reduce LTE network outages and improve availability, leading to universal 4G network maintenance through alarm analysis. The analyses of unsupervised learning methods demonstrated how ML solutions could improve network operations through better detection and remedy of network issues.

In another related study, Na et al. [10] employed a deep learning model, LSTM, to forecast future network performance. They gathered LTE network traffic data and transformed it using CUBIC and BBR trace log data, employing a sliding window approach to generate input data for the prediction model. This enhancement in the LSTM with the attention method demonstrated improved throughput prediction accuracy, which can be applied in time-sensitive applications to reduce latency. Yalda et al. [11] examined and described LSTM as a highly effective deep learning technique for tackling time series forecasting challenges. They trained the LSTM model using internet traffic data from an Internet service provider with three different optimisation algorithms to predict traffic flow. The results highlighted the effectiveness of the ReLU activation function and showed that the Adam optimiser outperformed other optimisation techniques, such as AdaGrad and stochastic gradient descent, in LSTM models for traffic prediction.

Furthermore, the applications of deep learning neural networks to resource utilisation in 4G radio access networks were proven in several research studies [12-14]. These practices highlighted the adaptability of deep neural networks in optimising resource management and which ultimately results in more effectual, consistent operations within 4G LTE networks.

A review of existing literature demonstrated that ML has garnered significant attention in the field of 4G LTE network management. This attention has been evidenced by the presentation of a number of promising solutions for capacity planning and effective network optimisation of the limited radio resources. However, modern network management is challenging due to factors including high traffic data variability, interference, environmental conditions, and subscriber behaviour [15]. Consequently, to ensure optimal resource allocation and QoS, prediction models must go through continuous enhancement.

## Materials And Methods

### Vanilla LSTM

Hochreiter and Schmidhuber developed the vanilla LSTM (vLSTM) model in 1997 to alleviate the problems affecting traditional recurrent neural networks (RNNs). By incorporating ‘gates’, which regulate the flow of information, LSTM uniquely enhanced its forecasting tasks involving sequences such as traffic patterns, speech recognition, and natural language modeling. These make it an optimal choice for this predictive task [16]. RNNs have memory components that enable them to store and retrieve information from preceding inputs to produce subsequent outputs in the sequence. Despite this capability, RNNs struggle with long-term dependencies and are prone to issues like vanishing and exploding gradients. They function by using an input at a time step $x_t$, a hidden state $h_{t-1}$, and an output $o_t$ at a specific time step [16]; these parameters are connected as follows:

$$x_t \in \mathbb{R}^{n_x},$$
$$h_t = \tanh\left(W_x x_t + W_h h_{t-1} + b_h\right),$$
$$o_t = W_o h_t + b_o.$$

Each unit in a vLSTM layer possesses memory capabilities, allowing it to retain information over extended periods. Three gates control information flow in the vLSTM through their combined operation to modify its cell state: input, forget, and output gates. The memory state $C_{t-1}$ acts as a memory that carries information across all cell components. The forget gate $f_t$ determines which cell gate information to keep or throw away while performing well to limit gradient disappearance along with explosion. This action considerably promotes the model's performance. $x_t$ is the input at the current time step, while $h_{t-1}$ is the previous hidden state of the cell. The sigmoid function $\sigma$ operates on $W_f$ weight and $b_f$ bias parameters. Based on the forget gate $f_t$ and during the operation, the input gate $i_t$ determines all new information, which needs to be stored or updated within the cell state. The new cell state $C_t$ updates itself with new information. Tangent hyperbolic activation function $(tanh)$ generates new information vector $\hat{C}$ through its output before adding it to the existing state. During training on the new cell state, the output gate requires $W_o$ weight parameters to determine the next hidden cell state based on the new cell state [17]. The network executes internal processes based on iterating the mathematical sequence below.

$$
\begin{aligned}
f_t &= \sigma\left(W_f[h_{t-1}, x_t] + b_f\right),\\
i_t &= \sigma\left(W_i[h_{t-1}, x_t] + b_i\right),\\
\tilde{C}_t &= \tanh\left(W_c[h_{t-1}, x_t] + b_c\right),\\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t,\\
o_t &= \sigma\left(W_o[h_{t-1}, x_t] + b_o\right),\\
h_t &= o_t \odot \tanh(C_t).
\end{aligned}
$$

**Stacked LSTM**

This involves the use of multiple LSTM layers, creating a stacked LSTM (sLSTM) model for traffic flow prediction. By stacking multiple LSTM layers, the model's capacity to express complex patterns increases, enhancing its capability to generalize and make accurate predictions [18]. This approach also mitigates the risk of overfitting to some extent. The sLSTM layers can assist in capturing and maintaining long-term dependencies in the data, which is essential for handling sequential data. Figure *2* illustrates the structure of the sLSTM model. In this study, the hidden state from the preceding layer is used as input for the ensuing layer, allowing the model to identify complex non-linear dependencies in the dataset. The output layer functions as the input for the subsequent attached layer.

**How to cite this article:**
Efuwape T O, Ojesanmi O A, Ogunbiyi T E, et al. (March 18, 2026) Enhancing Long-Term Evolution (LTE) Traffic Prediction Using Long Short-Term Memory Neural Networks. Cureus J Comput Sci 3 : es44389-025-00034-2. DOI https://doi.org/10.7759/s44389-025-00034-2

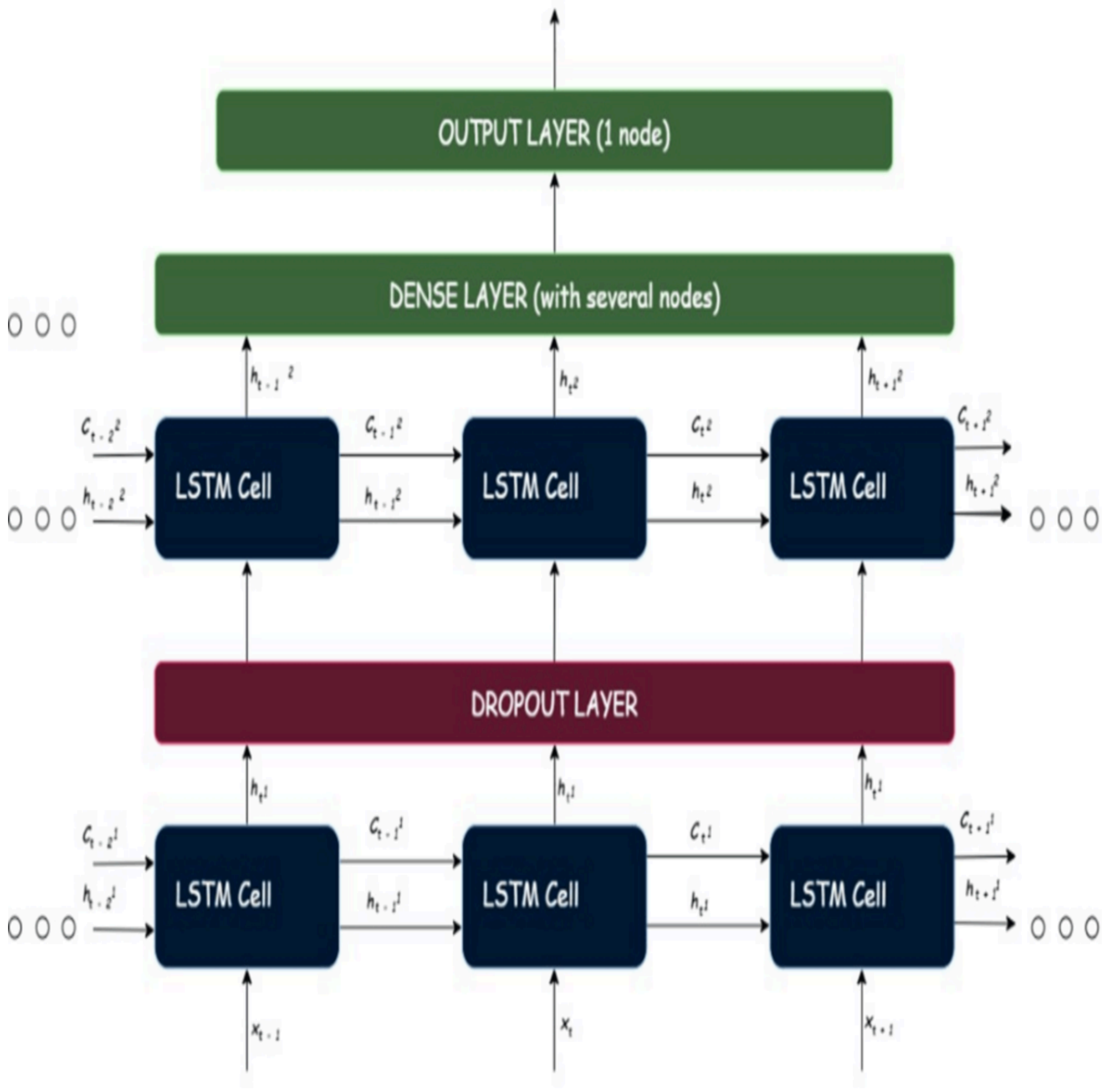


**FIGURE 2: Two-Layered Stacked LSTM**

LSTM, Long Short-Term Memory

### Bidirectional LSTM

The bidirectional LSTM (biLSTM) represents an enhanced version of the vLSTM, a sequence processing model comprising the forward and backward vLSTM algorithms. The integration of these two algorithms into a unified output layer [19] represents a significant advancement in the field of ML. The simultaneous operation of two distinct vLSTMs, processing the input data through their respective hidden states and linearly combining the two output results to produce the final output, with no connection between the two directions, is a key feature of biLSTM. This approach

**How to cite this article:**
Efuwape T O, Ojesanmi O A, Ogunbiyi T E, et al. (March 18, 2026) Enhancing Long-Term Evolution (LTE) Traffic Prediction Using Long Short-Term Memory Neural Networks. Cureus J Comput Sci 3 : es44389-025-00034-2. DOI https://doi.org/10.7759/s44389-025-00034-2

effectively increases the amount of information available to the algorithm. The final model takes into consideration the parameter factors of both past and future information, thereby enhancing the prediction algorithm's accuracy. The biLSTM equation can be expressed as follows:

$$f_t = \sigma\left(W_f[\overleftarrow{h}_{t+1}, x_t] + b_f\right),$$

$$i_t = \sigma\left(W_i[\overleftarrow{h}_{t+1}, x_t] + b_i\right),$$

$$\tilde{C}_t = \tanh\left(W_c[\overleftarrow{h}_{t+1}, x_t] + b_c\right),$$

$$C_t = f_t \odot C_{t+1} + i_t \odot \tilde{C}_t,$$

$$o_t = \sigma\left(W_o[\overleftarrow{h}_{t+1}, x_t] + b_o\right),$$

$$\overleftarrow{h}_t = o_t \odot \tanh(C_t).$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h}_t], \qquad \hat{y}_t = \sigma\left(Wh_t + b\right).$$

where $\hat{y}_t$ denotes the biLSTM model output, $\sigma(\cdot)$ is the sigmoid activation function, $\overrightarrow{h_t}$ and $\overleftarrow{h}_t$ represent the hidden-state outputs of the forward and backward LSTM networks, respectively, $W$ denotes the weight matrix, and $b$ is the bias term.

*Loss Functions*

The loss functions in ML are used to measure how well an algorithm can make accurate predictions on a given training dataset [20]. It is a mathematical function of the ML parameters, and they optimize these parameters during training to enhance accuracy, preventing overfitting. Choosing the wrong loss function affects the effectiveness of the algorithm. However, learning can be trained with different kinds of loss functions in different contexts to minimize the error between the expected and actual training values. This quantifies the differences and improves their decision-making assignment.

Mathematically, consider a collection of ($n$) data points, where $x_t$ denotes the actual (observed) value and $\hat{y}_t$ represents the predicted value at time step $t$. The loss functions employed in this study are designed to quantify the discrepancy between actual (observed )and predicted values and are defined as follows.

*Smooth_L1*

This loss function combines the properties and benefits of Huber loss. It aims to balance being too robust to outliers and less sensitive to small errors [21]. Often known as the smooth mean absolute error. The Smooth_L1 loss function is defined as follows:

$$L = \begin{cases} \frac{1}{2}\left(x_t - \hat{y}_t\right)^2, & \text{if } |x_t - \hat{y}_t| < 1, \\ |x_t - \hat{y}_t| - \frac{1}{2}, & \text{otherwise.} \end{cases}$$

*Mean Squared Error*

Mean squared error (MSE) represents an average between prediction values and real measurements through their squared difference value [22]. Because of taking the squared term, the evaluation method emphasizes large errors to deliver values that represent both the magnitude of change and error distributions. However, the value of MSE is uninterpretable. MSE is expressed as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n}\left(x_t - \hat{y}_t\right)^2$$

*Mean Squared Logarithmic Error*

Mean squared logarithmic error (MSLE) is a variant of MSE used to measure error in ML models. It calculates the ratio of the actual to modeled logarithmic value in the dataset [23]. MSLE's strength is that it can handle skewed data and different error sizes across values. MSLE is expressed as follows:

$$\mathrm{MSLE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{t}=1}^{\mathrm{n}} \left[\log\left(x_{\mathrm{t}}\right) - \log\left(\hat{y}_{\mathrm{t}}\right)\right]^2$$

*Root Mean Squared Error*

Root mean squared error (RMSE) determines the typical amount of deviations between predicted values and their actual counterparts. The mathematical calculation to derive RMSE requires obtaining the square root value from MSE measurements. A model's performance assessment relies on the RMSE value because lower values indicate superior accuracy by showing predictions closer to actual values [24-25]. This metric penalizes significant errors and is expressed as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{n}} \sum_{\mathrm{t}=1}^{\mathrm{n}} \left(x_{\mathrm{t}} - \hat{y}_{\mathrm{t}}\right)^2}$$

*Mean Absolute Error*

The mean absolute error (MAE) quantifies the average absolute difference between predicted and actual values. A smaller MAE indicates that the forecasts are closely aligned with the actual values, indicating a more effective model. MAE is particularly useful when the magnitude of prediction errors matters and the error distribution is not Gaussian [26]. However, compared to MSE, MAE is less sensitive to outliers. The MAE is calculated as follows:

$$\mathrm{MAE} = \frac{1}{n} \sum_{t=1}^{n} \left|x_t - \hat{y}_t\right|$$

*Huber Loss*

Huber loss combines the best properties of two mean errors, the squared error (MSE) and the absolute error (MAE), while at the same time maintaining smoothness and preventing the gradient from descending too fast and missing the prime target [27]. The Huber loss is quadratic and hybrid in nature, making it less sensitive to outliers outside the threshold (δ) but also penalizing small errors within the data points, similar to MSE. However, computation cost is high. The Huber loss function is expressed as follows:

$$L(x_t, \hat{y}_t) = \begin{cases} \frac{1}{2}(x_t - \hat{y}_t)^2, & \text{if } |x_t - \hat{y}_t| \le \delta, \\ \delta\left(|x_t - \hat{y}_t| - \frac{1}{2}\delta\right), & \text{otherwise,} \end{cases}$$

where $\delta > 0$ is a predefined threshold that controls the transition between quadratic and linear loss.

*Log-Cosh*

This refers to the logarithm of the hyperbolic cosine of the error. It behaves similarly to Huber loss but is smoother and does not have a critical threshold [28]. For small errors, it acts like MSE, and for large errors, it behaves like MAE, which helps enhance training stability. However, it is less adaptive since it follows a static scale. The Log-Cosh loss is expressed as follows:

$$\mathcal{L}(x_t, \hat{y}_t) = \frac{1}{n} \sum_{t=1}^{n} \log\left(\cosh\left(x_t - \hat{y}_t\right)\right)$$

*Quantile Loss*

This modifies the weight of each test value based on the desired quantile and is used to predict a range of values rather than a single value [28]. Overestimation or underestimation can occur when a model’s prediction is higher or lower than the actual value. It is most often applied in quantile regression, where the quantile lies between 0 and 1.

$$L_\tau(x_t, \hat{y}_t) = \begin{cases} \tau(x_t - \hat{y}_t), & \text{if } x_t \ge \hat{y}_t, \\ (1-\tau)(\hat{y}_t - x_t), & \text{otherwise,} \end{cases} \text{where } \tau \in (0,1) \text{ represents the target quantile level.}$$

**Dataset characteristics and preprocessing**

The dataset used in this study was obtained from an operational 4G LTE cellular network site located in Sub-Saharan Africa. The data were collected through the Huawei U2020 network monitoring and statistics server and consist of key performance indicators recorded for each LTE cell. The recorded attributes include eNodeB identification, CellID, local cell ID, geographical coordinates, cluster information, timestamp, uplink and downlink traffic volumes, and uplink and downlink physical resource block utilization. The overall analytical procedure follows the Cross-Industry Standard Process for Data Mining framework [29].

To support temporal analysis and reduce short-term traffic fluctuations, the raw network records were aggregated to daily observations. Data aggregation was performed using the CellID and time-related fields, which further allow traffic and resource utilization metrics to be summarized on per-day basis. The step provides a stable representation of network behavior while retaining essential temporal characteristics.

Since incomplete records can negatively affect ML performance, missing values present in the dataset were handled prior to model training. Statistical imputation was applied. Depending on the statistical distribution of each variable, missing entries were replaced using either the mean or median value computed from the available data. Consequently, this approach ensures data completeness without significantly altering the underlying data distribution.

Before proceeding to model development, the data were normalized using Min-Max scaling to transform all values into the range [0;1]. Normalization reduces scale differences and improves numerical stability during model training.

**Hyperparameter optimization**

A systematic hyperparameter tuning process was conducted using feedback from the validation dataset. Key model parameters including the quantity of LSTM units, dropout rates, subset of the training data and weights, and the complete pass size were carefully adjusted to enhance the network's learning and generalization capabilities. This would improve the predictive accuracy of the three LSTM-based models.

The hyperparameters for all three LSTM variants were ultimately selected through a grid search approach. This method evaluates a predefined combination of parameter values, generating a structured set of configurations that allows the model to achieve its best possible performance. By systematically discovering this parameter space, the grid search minimizes reliance on manual trial-and-error adjustments, thereby enhancing efficiency and optimally reducing the potential for human error.

The range of hyperparameters considered in the tuning process significantly influenced model performance, as shown in Table *1*. For instance, varying the quantity of LSTM units affected the network capacity to capture temporal dependencies in LTE traffic data, while different dropout rates helped control overfitting. Similarly, subset of the training data and learning weights influenced the convergence speed and stability during training. Systematically, discovering these ranges enabled identification of parameter combinations that optimized both the accuracy and efficiency of the LSTM-based models.

| Hyperparameter | Value |
| --- | --- |
| Activation function (input layer) | Tangent hyperbolic |
| Activation function (output layer) | Sigmoid |
| Loss functions | MSE, MAE, Smooth_L1, RMSE, Huber loss, Quantile, Log-Cosh, MSLE |
| Learning rate | 0.0001 |
| Batch size | 32 |
| Number of epochs | 100 |
| Train set | 26,879 |
| Test set | 6,720 |
| Optimizer | Adam |
| Dropout rate | 0.5 |

**TABLE 1: Hyperparameter Optimization for LTE Traffic Prediction**

LTE – Long-Term Evolution; MSE – Mean Squared Error; MAE – Mean Absolute Error; MSLE – Mean Squared Logarithmic Error; RMSE – Root Mean Squared Error

### Experimental setup

Python was selected due to its wide adoption in scientific computing, ML, and data analytics, as well as its extensive ecosystem of open-source libraries. Model development and experimentation were conducted using Google Collaboratory, with additional support from standard Python libraries including Matplotlib for visualization, Scikit-learn for preprocessing and evaluation utilities, and TensorFlow (version 2.3.1) for deep learning implementation. TensorFlow provides a flexible and efficient framework for designing, training, and optimizing neural network architectures. The LSTM-based models were implemented using the Python programming language (version 3.8).

Model training was carried out through a structured process to ensure effective learning from historical traffic data and reliable generalization to unseen samples. The LSTM models were trained using a batch size of 32 over 100 epochs, resulting in a total of 26,879 trainable parameters. The learning rate was fixed at 0.0001, with parameter updates applied at regular intervals of 30 steps. The network architecture employs the hyperbolic tangent function for state transformations and the sigmoid function for gating mechanisms, which are standard choices for LSTM-based sequence modeling.

In ensuring robust performance, the models were evaluated under different loss function configurations, while the Adaptive Moment Estimation (Adam) optimizer was used for weight updates due to its effectiveness in handling stochastic objectives and sparse gradients. Model performance and convergence behavior were monitored and analyzed through graphical visualization using the Matplotlib library.

**How to cite this article:**
Efuwape T O, Ojesanmi O A, Ogunbiyi T E, et al. (March 18, 2026) Enhancing Long-Term Evolution (LTE) Traffic Prediction Using Long Short-Term Memory Neural Networks. Cureus J Comput Sci 3 : es44389-025-00034-2. DOI https://doi.org/10.7759/s44389-025-00034-2

### LTE traffic prediction and model evaluation

Normalized dataset was partitioned into train and test subsets using an 80-20 split. The training subset was employed to learn temporal dependencies and traffic variations using vanilla, stacked, and biLSTM architectures. The remaining portion of the data was reserved for model evaluation to assess predictive performance on unseen observations. This separation ensures that model assessment reflects generalization capability rather than memorization of historical patterns.

Model performance was evaluated using established quantitative metrics commonly applied in time-series forecasting and network traffic analysis. Specifically, MSE and the coefficient of determination, also denoted as $R^2$, were used to measure prediction accuracy and goodness of fit, respectively. These metrics provide objective criteria for comparing model outputs with observed traffic values.

The error-based metric (MSE) quantifies the deviation between predicted and actual (observed) traffic values. It penalizes larger prediction errors more heavily through quadratic weighting, making it sensitive to substantial deviations in traffic demand. The $R^2$ metric complements this analysis by indicating the proportion of variance in downlink traffic that are accounted by the predictive models. Together, these evaluation measures provide a balanced assessment of both accuracy and explanatory performance, supporting reliable validation of the LTE traffic prediction framework.

## Results

| Loss Functions | vLSTM | sLSTM | biLSTM |
|---|---|---|---|
| Huber loss | 0.002674 | 0.003234 | 0.002925 |
| Log_Cosh | 0.003412 | 0.002491 | 0.002562 |
| MSE | 0.002239 | 0.002406 | 0.695326 |
| MAE | 0.004229 | 0.002385 | 0.002760 |
| MSLE | 0.002268 | 0.002388 | 0.002871 |
| QP | 0.002655 | 0.002254 | 0.002406 |
| Smooth_L1 | 0.022586 | 0.003015 | 0.002759 |
| RMSE | 0.002262 | 0.002297 | 0.002554 |

**TABLE 2: Error Values of the Prediction Models**

MSE – Mean Squared Error; MAE – Mean Absolute Error; MSLE – Mean Squared Logarithmic Error; RMSE – Root Mean Squared Error; vLSTM – Vanilla Long Short-Term Memory; sLSTM – Stacked Long Short-Term Memory; biLSTM – Bidirectional Long Short-Term Memory

| Loss Functions | vLSTM | sLSTM | biLSTM |
|---|---|---|---|
| Huber loss | 0.838073 | 0.804122 | 0.822856 |
| Log_Cosh | 0.793362 | 0.849156 | 0.844827 |
| MSE | 0.8644300 | 0.855578 | 0.832832 |
| MAE | 0.743885 | 0.854290 | -41.109460 |
| MSLE | 0.862667 | 0.855410 | 0.826138 |
| QP | 0.839229 | 0.863497 | 0.832923 |
| Smooth_L1 | 0.843389 | 0.817400 | 0.854319 |
| RMSE | 0.862994 | 0.860888 | 0.845347 |

**TABLE 3: Summary of the Performance Accuracy**

MSE – Mean Squared Error; MAE – Mean Absolute Error; MSLE – Mean Squared Logarithmic Error; RMSE – Root Mean Squared Error; vLSTM – Vanilla Long Short-Term Memory; sLSTM – Stacked Long Short-Term Memory; biLSTM – Bidirectional Long Short-Term Memory

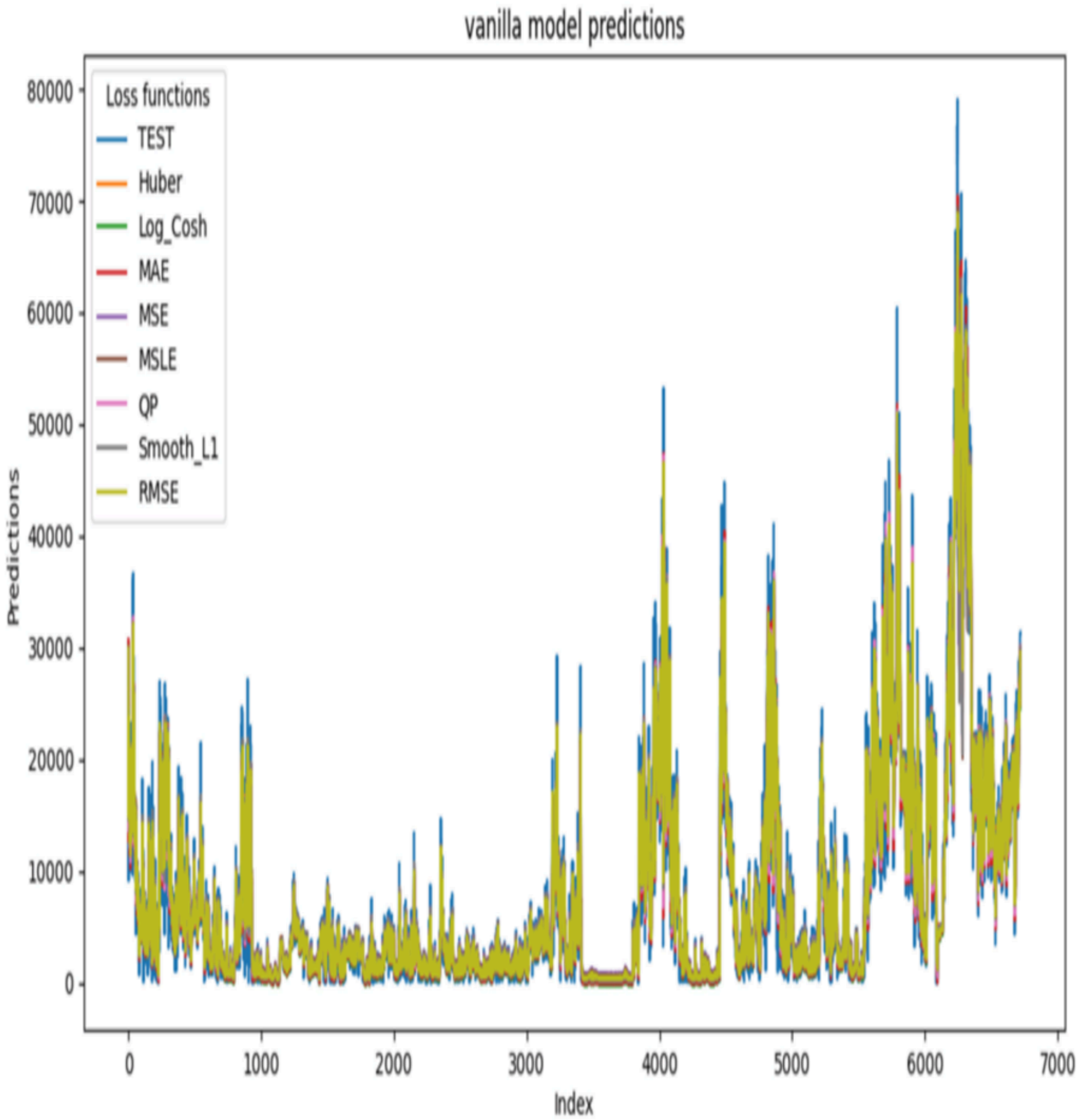


**FIGURE 3: Validation Under Different Loss Functions - Vanilla LSTM**

MSE – Mean Squared Error; MAE – Mean Absolute Error; MSLE – Mean Squared Logarithmic Error; RMSE – Root Mean Squared Error; LSTM – Long Short-Term Memory

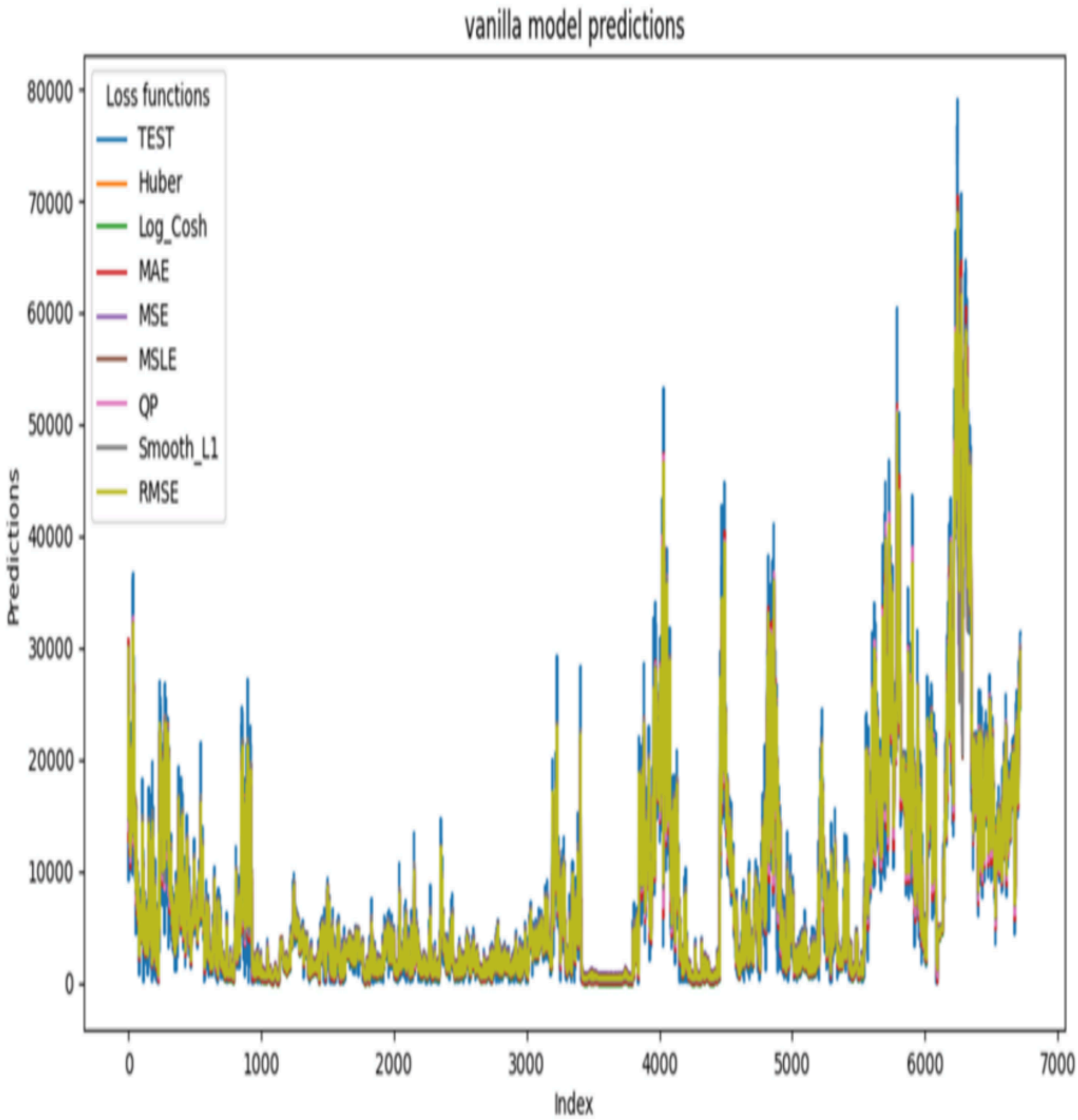


**FIGURE 4: Validation Under Different Loss Functions - Stacked LSTM**

MSE – Mean Squared Error; MAE – Mean Absolute Error; MSLE – Mean Squared Logarithmic Error; RMSE – Root Mean Squared Error; LSTM – Long Short-Term Memory

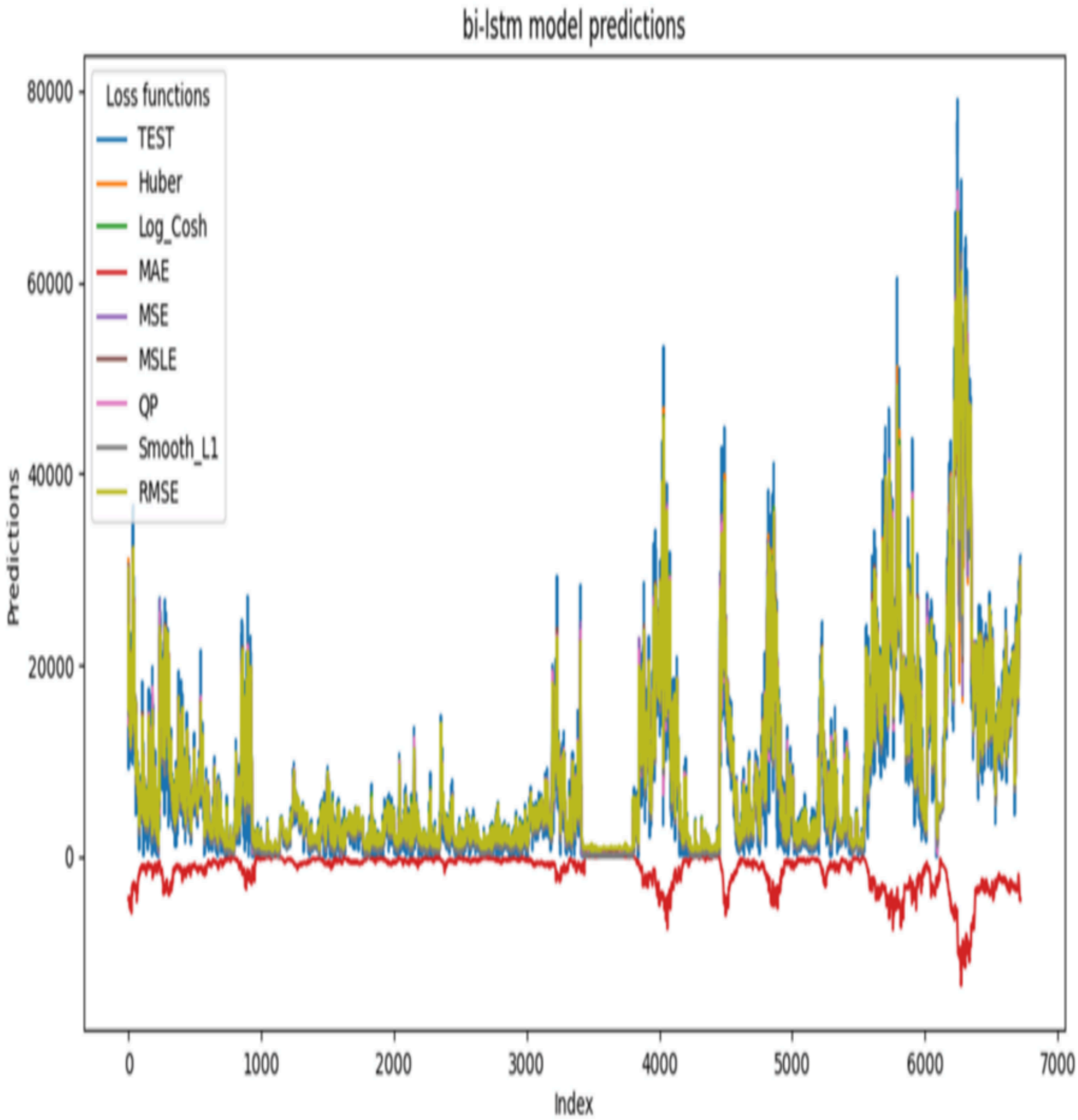


**FIGURE 5: Validation Under Different Loss functions - Bidirectional LSTM**

MSE – Mean Squared Error; MAE – Mean Absolute Error; MSLE – Mean Squared Logarithmic Error; RMSE – Root Mean Squared Error; LSTM – Long Short-Term Memory

## Discussion

The three LSTM models' predictive performance is shown by the error values in Table 2, which are based on test values derived from multiple loss functions. The model's predictive ability improves as the error value approaches zero. A case of a higher error value of 0.695326 for biLSTM with the MAE loss function indicated the weakest predictive performance.

**How to cite this article:**
Efuwape T O, Ojesanmi O A, Ogunbiyi T E, et al. (March 18, 2026) Enhancing Long-Term Evolution (LTE) Traffic Prediction Using Long Short-Term Memory Neural Networks. Cureus J Comput Sci 3 : es44389-025-00034-2. DOI https://doi.org/10.7759/s44389-025-00034-2

For each model, Table 3 provides exact quantitative results for prediction accuracy with respect to various loss functions. The LTE traffic prediction achieved 86.44% accuracy when applying the vLSTM model with the MSE loss function, but loss accuracy decreased to 86.35% when stacking the LSTM with the quantile loss function. While a negative prediction accuracy (-41.109546) for the MAE loss function is shown, the biLSTM achieves 85.43% accuracy during learning using the Smooth_L1 loss function. Each LSTM model was run several times during network traffic prediction in order to guarantee the reliability of the training results. Hence, it produced consistent and stable results with little variation between subsequent executions. The Adam optimizer contributed to stability and generated different prediction results because its feature adjusts model parameters by adapting learning rates based on past update directions and sizes. The lower predicted and actual value differences stem from this account.

Figures 3-5 compare the observed and the predicted (test) values for the three LSTM models. It provides validation for the various loss functions employed in this study. Both the sLSTM and the vLSTM perform well across a range of loss functions, according to the graphs. This implies that these models are well equipped for predicting LTE traffic. With the exception of using the MAE loss function, as shown by the red color, biLSTM typically performs well during learning in Figure 5. It is evident that there was a significant difference between the test and predicted values for the bidirectional LSTM. This observation implies that the biLSTM model might not be the most suitable choice for the LTE traffic demand prediction.

## Conclusions

This study examined the effectiveness of deep learning models in LTE traffic demand prediction using vLSTM, sLSTM, and biLSTM, as well as various loss functions to enhance optimization. Every loss function examined in this study has unique properties that make them essential for accelerating parameter optimization during the training phase. The loss functions work by gradually changing the model's parameters, which simultaneously lowers prediction errors relative to actual values and raises the accuracy level of the model. The study determines the optimal loss functions for LTE traffic demand prediction using the coefficient of determination $R^2$ and MSE metrics. The research also illustrates how accurate predictions and convergence worked out for training and testing datasets throughout the evaluation of three LSTM-based algorithms. The investigated results identify how different loss functions affect the performance of each algorithm, respectively. Deep learning requires a fundamental understanding of loss function differences to select suitable ones because this choice directly affects both learning model operation and training dynamics. LSTMs maintain essential information throughout long durations, thereby delivering accurate forecasts in scenarios where historical pattern data determines future activities. LSTM neural networks enable successful future traffic projection through their capability to identify long-term traffic pattern connections, which makes them an ideal platform for this prediction purpose.

Future performance improvements could be attained by using time-series augmentation techniques to increase variability and expand the dataset. In addition, transfer learning approaches leveraging traffic data from comparable network environments can help mitigate data scarcity. The integration of supplementary contextual features, such as user mobility patterns, quality of service indicators, or external demand factors, could further improve model robustness and enhance predictive reliability under diverse operating conditions.

## Additional Information

### Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

**Concept and design:** Temitope O. Efuwape, Omolara O. Akinlade, Olusegun A. Ojesanmi, Temitope E. Ogunbiyi

**Acquisition, analysis, or interpretation of data:** Temitope O. Efuwape, Omolara O. Akinlade, Olusegun A. Ojesanmi, Temitope E. Ogunbiyi

**Drafting of the manuscript:** Temitope O. Efuwape, Omolara O. Akinlade, Olusegun A. Ojesanmi, Temitope E. Ogunbiyi

**Critical review of the manuscript for important intellectual content:** Temitope O. Efuwape, Omolara O. Akinlade, Olusegun A. Ojesanmi, Temitope E. Ogunbiyi

**Supervision:** Olusegun A. Ojesanmi, Temitope E. Ogunbiyi

### Disclosures

**Human subjects:** All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

## References


1. Wu D, Li J, Ferini A, et al.: Reinforcement learning for communication load balancing: approaches and challenges. Frontiers in Computer Science. 2023, 5:1156064. 10.3389/fcomp.2023.1156064
2. Ericsson Mobility Report: Mobile data traffic outlook. (2023). Accessed: June 30, 2023: https://d110erj175o600.cloudfront.net/wp-content/uploads/2023/06/21123003/Ericsson-Mobility-Report-June-2023.pdf.
3. Hussein SA, Répás SR: Enhancing network security through machine learning-based anomaly detection systems. International Journal of Intelligent Systems and Applications in Engineering. 2024, 12:1929-1935.
4. Schummer P, del Rio A, Serrano J, Jimenez D, Sánchez G, Llorente Á: Machine learning-based network anomaly detection: Design, implementation, and evaluation. AI. 2024, 5:2967-2983. 10.3390/ai5040143
5. Chen A, Law J, Aibin M: A survey on traffic prediction techniques using artificial intelligence for communication networks. Telecom. 2021, 2:518-535. 10.3390/telecom2040029
6. Wang X, Wang Z, Yang K, Song Z, Bian C, Feng J, Deng C: A survey on deep learning for cellular traffic prediction. Intelligent Computing. 2024, 3:0054. 10.34133/icomputing.0054
7. Alsaade FW, Al-Adhaileh MH: Cellular traffic prediction based on an intelligent model. Mobile Information Systems. 2021, 2021:6050627. 10.1155/2021/6050627
8. Kuboye BM, Adedipe AI, Oloja SV, Obolo OA: Users' evaluation of traffic congestion in LTE networks using machine learning techniques. Artificial Intelligence Advances. 2023, 5:8-24. 10.30564/aia.v5i1.5452
9. Bernabe B, Michel DDE, Marie CA, Fabrice MTS: Comparing machine learning algorithms for improving the maintenance of LTE networks based on alarms analysis. Journal of Computer and Communications. 2022, 10:125-137. 10.4236/jcc.2022.1012010
10. Na H, Shin Y, Lee D, Lee J: LSTM-based throughput prediction for LTE networks. ICT Express. 2023, 9:247-252. 10.1016/j.icte.2021.12.001
11. Yalda K, Hamad DJ, Tapus N, Okumus IT: Network traffic prediction performance using LSTM. Romanian Journal of Information Science and Technology. 2024, 27:336-347.
12. Djomadji E, Kabiena I, Nkemeni V, Belinga À Njere A, Sone M: Dynamic resource allocation in LTE radio access network using machine learning techniques. Journal of Computer and Communications. 2023, 11:73-93. 10.4236/jcc.2023.116005
13. Rezazadeh F, Zanzi L, Devoti F, Barrachina-Muñoz S, Zeydan E, Costa-Pérez X, Mangues-Bafalluy J: A multi-agent deep reinforcement learning approach for RAN resource allocation in O-RAN. IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hoboken, NJ, USA. 2023, 1-2. 10.1109/INFOCOMWKSHPS57453.2023.10226154
14. Qazzaz MMH, Kułacz Ł, Kliks A, Zaidi SA, Dryjanski M, McLernon D: Machine learning-based xApp for dynamic resource allocation in O-RAN networks. 2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN), Stockholm, Sweden. 2024, 10:492-497. 10.1109/ICMLCN59089.2024.10625184

15. Yuliana H., Hendrawan I, Musashi Y: Estimating base station traffic and throughput using machine learning based on hourly key performance indicator (KPI) network analysis. IEEE Access. 2024, 12:116285-116301. 10.1109/access.2024.3447098
16. Efuwape TO, Ojesanmi OA: A conceptual framework of eNodeB peak patterns prediction based dynamic pricing scheme for enhanced quality of LTE network services. International Journal of Information Resource Management. 2025, 2:1-21.
17. Zhang A, Lipton ZC, Li M, Smola AJ: Dive Into Deep Learning. Cambridge University Press, Cambridge, UK; 2024.
18. Li K, Bai W, Huang S, Tan G, Zhou T, Li K: Lag-related noise shrinkage stacked LSTM network for short-term traffic flow forecasting. IET Intelligent Transport Systems. 2024, 18:244-257. 10.1049/itr2.12448
19. Zhang W, Xue H, Jiang X, Liu J: An optimized bidirectional long short-term memory model based on hyperspectral analysis of protein content in milk powder. Food Science & Nutrition. 2024, 13:e4556-. 10.1002/fsn3.4556
20. Terven J, Cordova-Esparza DM, Romero-González JA, Ramírez-Pedraza A, Chávez-Urbiola EA: A comprehensive survey of loss functions and metrics in deep learning. Artificial Intelligence Review. 2025, 58:195. 10.1007/s10462-025-11198-7
21. Wang Q, Ma Y, Zhao K, Tian Y: A comprehensive survey of loss functions in machine learning. Annals of Data Science. 2022, 9:187-212. 10.1007/s40745-020-00253-5
22. Hodson TO, Over TM, Foks SS: Mean squared error, deconstructed. Journal of Advances in Modeling Earth Systems. 2021, 13:e2021MS002681. 10.1029/2021ms002681
23. Park Y: Concise logarithmic loss function for robust training of anomaly detection model. arXiv. 2023, 10.48550/arXiv.2201.05748
24. Van Poecke A, Meuris L, Cisneros M, Van Ginderachter M, Hellinckx P, Tabari H: A comparative sensitivity analysis of loss functions in machine learning-based weather forecasting. Advances on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC 2024. Lecture Notes on Data Engineering and Communications Technologies. Barolli L (ed): Springer, Cham; 2024. 232:318-326. 10.1007/978-3-031-76462-2_29
25. Vos PW: Rethinking mean square error: why information is a superior assessment of estimators. arXiv. 2024, 10.48550/arXiv.2412.08475
26. Robeson SM, Willmott CJ: Decomposition of the mean absolute error (MAE) into systematic and unsystematic components. PLoS ONE. 2023, 18:e0279774. 10.1371/journal.pone.0279774
27. Malekzadeh P, Plataniotis KN, Poulos Z, Wang Z: A robust quantile huber loss with interpretable parameter adjustment in distributional reinforcement learning. ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea. 2024, 6120-6124. 10.1109/ICASSP48485.2024.10447501
28. Noel MM, Banerjee A, Oswal Y, Amali DGB, Muthiah-Nakarajan V: Alternate loss functions for classification and robust regression can improve the accuracy of artificial neural networks. arXiv. 2023, 10.48550/arXiv.2303.09935
29. Schröer C, Kruse F, Gómez JM: A systematic literature review on applying CRISP-DM process model. Procedia Computer Science. 2021, 181:526-534. 10.1016/j.procs.2021.01.199


**How to cite this article:**
Efuwape T O, Ojesanmi O A, Ogunbiyi T E, et al. (March 18, 2026) Enhancing Long-Term Evolution (LTE) Traffic Prediction Using Long Short-Term Memory Neural Networks. Cureus J Comput Sci 3 : es44389-025-00034-2. DOI https://doi.org/10.7759/s44389-025-00034-2