Improving Cellular Traffic Prediction with Temporal Embeddings: A Time2Vec-LSTM Approach

Hamidullah Riaz*, ‡, Sıtkı Öztürk**, Peri Güneş***

*, ** Department of Electronics and Communication, Faculty of Engineering, Kocaeli University, Izmit, Kocaeli, 41001, Türkiye.

*** Department of Electrical and Electronics, Faculty of Engineering and Architecture, Istanbul Gelisim University, Avcılar, Istanbul, 34310, Türkiye.

(hamidullah.riaz@kocaeli.edu.tr, sozturk@kocaeli.edu.tr, pgunes@gelisim.edu.tr)

(ORCID: 0000-0001-5275-9922 ; 0000-0003-3804-5581 ; 0009-0002-9080-3239)

‡ Corresponding Author; Kocaeli University, Izmit, Kocaeli, 41001, Türkiye, hamidullah.riaz@kocaeli.edu.tr

Received: 08.05.2025 Accepted: 08.08.2025

Abstract- The network traffic prediction has to be reliable for better resource allocation and congestion management in present-day telecommunications. In this paper, a novel hybrid Time2Vec-enhanced LSTM method is presented for somewhat more accurate traffic volume forecasting. The model exploits both historical traffic behavior and temporal features enriched by Time2Vec, such as hour and day, to represent the linear or periodic dependencies embedded in cellular traffic. Unlike traditional static time encodings or raw time features, the learnable Time2Vec embeddings enable the model to better capture daily and hourly fluctuations in network traffic. The study carried out experiments with a real-world dataset that had been collected from an LTE base station located in Kandahar Province of Afghanistan, with hourly uplink, downlink, and total traffic volumes recorded for 30 days. Performance was measured in terms of the Root Mean Square Error (RMSE) and coefficient of determination (R^2). The results show that the proposed Time2Vec-enhanced LSTM consistently outperforms Deep Learning (DL), statistical, and Machine Learning (ML) models across all traffic types. The learnable temporal embeddings are useful as they allow greater accuracy and better capture of trends. Ablation studies have supported that forecasting is far better with adaptive Time2Vec encoding than with models without or with a fixed-time feature, suggesting that learnable temporal features are essential for precise and robust cellular traffic prediction.

Keywords- Cellular traffic prediction, Deep learning (DL), LSTM, Machine learning (ML), Real-word dataset, Time2Vec.

1. Introduction

The rapid growth of internet and mobile technologies, along with the increasing use of smartphones and other connected devices, has brought about a new era of big data. This has led to a sharp rise in global mobile data usage, which is expected to reach 403 exabytes (EB) per month by 2029 [1]. The growing number of users, applications, and services is putting pressure on communication networks. By the end of 2023, mobile data traffic, excluding fixed wireless access, was estimated at 130 EB per month, with forecasts suggesting this could climb to 563 EB per month by 2029 when fixed wireless is included. At the same time, 5G networks were predicted to carry 25% of this traffic by the end of 2023, increasing significantly to 76% by 2029 [2], [3]. These trends highlight the urgent need for accurate forecasting methods to support better planning, traffic control, and efficient use of resources, while also improving the quality of service for users.

As mobile data usage continues to increase, managing network capacity effectively becomes more difficult. One key solution is accurate prediction of cellular network traffic, which supports better network operations. Being able to anticipate traffic patterns can help avoid congestion, improve security, and guide the efficient distribution of bandwidth. It also plays a central role in long-term planning by allowing network providers to prepare for future demands. With reliable forecasting, service providers can make more informed decisions and optimize their infrastructure to deliver a smoother and more secure user experience [4].

Cellular traffic prediction is generally divided into temporal and spatiotemporal approaches. Temporal prediction focuses on forecasting traffic at a single location using only its historical data, typically modeled as a univariate time series. In contrast, spatiotemporal prediction captures both temporal patterns and spatial dependencies across network elements, for example base stations, which are influenced by factors such as user mobility and handovers. Based on the number of predicted variables, traffic forecasting can be classified as univariate or multivariate. Univariate methods predict a single variable, such as traffic volume, while multivariate methods forecast multiple related indicators, such as traffic volume and the number of connected users, which often influence one another. Forecasting is also categorized by duration, with short-term prediction typically covering 5 to 60 minutes and medium-to-long-term prediction extending beyond 60 minutes, depending on the data's time granularity. Additionally, predictions may be single-step, focusing on the next time point, or multi-step, predicting traffic over several future intervals [5], [6].

However, despite advances in Machine Learning (ML) and Deep Learning (DL), many existing models still rely on static or hand-engineered time features, which limits their ability to adapt to the complex, variant nature of cellular traffic. To address this limitation, we explore the use of learnable temporal embeddings as a more flexible and data-driven alternative to traditional time encodings.

The aim of the study is to incorporate the Time2Vec mechanism within an LSTM-based architecture for cellular traffic forecasting. Time2Vec is a learnable temporal embedding that represents time as a vector with linear and periodic components. Unlike static encodings, Time2Vec allows the model to automatically discover and adapt to recurring and drifting temporal patterns within the data. To the best of our knowledge, this is the first study to apply Time2Vec in the context of cellular traffic prediction.

Our Time2Vec-enhanced LSTM architecture, by embedding temporal information in a manner that can be learned simultaneously with traffic patterns, will thus provide better temporal awareness and more accurate predictions. We in turn validate this model on actual LTE traffic data collected from a live cellular network, demonstrating that it consistently outperforms a diverse range of methods including DL approaches such as LSTM and GRU, traditional time series model such as Auto-Regressive Integrated Moving Average (ARIMA), and classical ML techniques such as Support Vector Regression (SVR) and Random Forest (RF). This highlights the potential of combining sequence modeling with a powerful temporal embedding to improve the robustness and accuracy of cellular traffic forecasting.

The key contributions of this work are as listed below:

- We propose a hybrid Time2Vec-enhanced LSTM model where learnable temporal embeddings are incorporated for sequential modeling to gain higher accuracy in cellular traffic forecasting.
- We introduce Time2Vec into the telecom traffic prediction field, demonstrating that it offers superior performance compared to traditional static time encodings or raw time features, particularly for modeling daily and hourly fluctuations in network traffic.
- We validate the model on actual LTE traffic data collected from a base station in Afghanistan, featuring uplink, downlink, and total traffic volume, all recorded hourly over 30 days.
- We conduct comprehensive empirical comparisons with DL models (LSTM and GRU), traditional statistical (ARIMA), and ML models (SVR and RF), demonstrating that our proposed model consistently achieves lower Root Mean Square Error (RMSE) and higher coefficient of determination (R2) across all traffic types.

The remainder of the paper is organized as follows: Section 2, Section 3 describes the Materials and Methods, including the dataset, LSTM architecture, and Time2Vec embedding. Section 3.3 presents the Proposed Model. Section 4 discusses the Results and Analysis, and finally, Section concludes the paper.

2. Related Work

The task of forecasting traffic in cellular networks has traditionally been addressed using statistical time-series models such as ARIMA [7] and Seasonal ARIMA (SARIMA) [8]. These models are valued for their mathematical simplicity and effectiveness in environments with stable and predictable patterns. In particular, they tend to perform reasonably well in short-term forecasting scenarios. However, their inherent limitations become evident when applied to the highly dynamic and complex nature of modern cellular networks. Real-world traffic patterns are often nonlinear, influenced by diverse user behaviors, mobility patterns, and spatial interdependencies between network nodes. Linear models, by design, are not well suited to represent such complexity. Their reliance on assumptions of stable statistical properties over time and their limited capacity to model long-range or spatial correlations result in reduced predictive accuracy and reliability, particularly in long-term forecasting tasks. Consequently, while these methods remain useful in controlled settings, their applicability to large-scale, real-world networks is significantly constrained.

The continuous growth of network traffic, along with recent developments in ML, has led to increased interest in data-driven approaches for cellular traffic prediction. These methods are viewed as promising alternatives to traditional statistical models, particularly in handling the complexity and variability of modern network environments. However, simpler ML algorithms such as linear regression and support vector regression, often fall short. Their limited capacity to capture nonlinear and high-dimensional patterns makes them less suitable for accurate forecasting in real-world scenarios.

To overcome these limitations, researchers have increasingly adopted advanced DL architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which are specifically designed to handle long-term dependencies. These models have been widely used for cellular traffic prediction due to their ability to learn from large volumes of sequential data [9], [10], [11]. For example, an LSTM-based traffic prediction model utilizing real-world call data is proposed in [12], demonstrating its ability to learn meaningful patterns in practical scenarios. Similarly, in [13], an LSTM model is used to predict a retainability Key Performance Indicator (KPI) from Ericsson's Long-Term Evolution (LTE) network in Algeria. On the other hand, GRUs, which offer a more computationally efficient alternative to LSTMs, have also been explored for similar applications [14]. Additionally, [15] proposes a GRU-based neural network model that predicts base station traffic by capturing the periodicity and fluctuating characteristics inherent in base station traffic data. Moreover, hybrid models that integrate LSTM or GRU with Convolutional Neural Networks (CNNs) have shown potential for improved feature extraction, though they come with added complexity and computational overhead [16], [17].

In general, recurrent models such as LSTM and GRU are considered excellent choices when it comes to modeling sequential patterns in cellular traffic data; however, their way of representing and leveraging any time-related information is limited. Standard approaches make use of static features for time representation, such as hour-of-day or cyclical encodings of time-not nearly sufficient to capture the richness of temporal patterns, such as long-term seasonal trends or the subtle finesse of daily and weekly cycles or demand surges at certain hours. The absence of a trainable expressive time representation renders a model incapable of adapting to non-stationary and multi-scale traffic dynamics.

3. Methodology

This section outlines the components of the proposed Time2Vec-enhanced LSTM model, including the Time2Vec encoding technique, the LSTM architecture, integration of Time2Vec, and the model's input-output structure, along with the training approach.

3.1. Time2Vec Embedding

Time2Vec is a time-encoding method that effectively incorporates temporal data into ML models. Rather than requiring manual construction of time-based features, Time2Vec learns a representation that includes long-term trends and recurring patterns in time-series data [18], [19]. Such an approach is very helpful in traffic prediction tasks, where time-dependent behaviors are either periodic (daily, weekly, etc.) or changing across time. For every scalar input t, Time2Vec produces a vector of size $k+1$, where the first component defines a linear transformation of time (modeling aperiodic trends), whereas the remaining k components model periodic variations in time using sinusoidal functions with trainable parameters for frequency and phase. This empowers Time2Vec to learn some of the complex time-related dynamics that are necessary in various real applications.

The Time2Vec function is mathematically expressed as Eq. (1):

$$Time2Vec = \begin{cases} \omega_0 \cdot t + b_0 & \text{if } i = 0 \\ \sin(\omega_i \cdot t + b_i) & \text{for } 1 \leq i \leq k \end{cases} \tag{1}$$

where ω_i and b_i are trainable weights and biases. The linear component $\omega_0 \cdot t + b_0$ captures long-term, non-periodic trends, while the sinusoidal components $\sin(\omega_i \cdot t + b_i)$ model periodic behaviors such as daily or weekly cycles.

3.2. LSTM Architecture Overview

LSTM is a type of Recurrent Neural Network (RNN) aimed at addressing the limits of traditional RNNs in modeling long-range dependencies within sequential data. This enables LSTM to excel in time series forecasting by retaining memory for long time intervals and avoiding issues such as vanishing gradients. The LSTM architecture consists mainly of four key components: the memory cell, the forget gate, the input gate, and the output gate, as illustrated in Fig. 1. The memory cell is a unit maintaining information and allows information to be continuously fed into following time steps, while the gates regulate what to allow in or out of the memory cell and what should be forgotten.

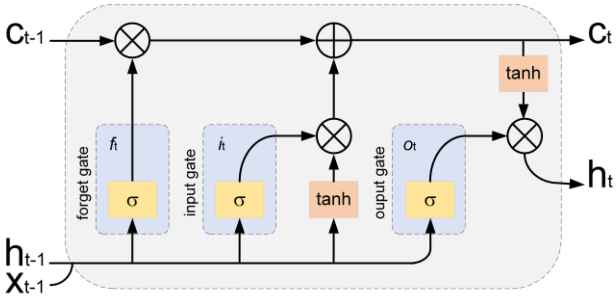

Fig. 1: LSTM cell structure showing the forget, input, and output gates, and the update of cell and hidden states

The forget gate is defined as Eq. (2):

$$f_t = \sigma\left(\omega_f [h_{t-1}, x_t] + b_f\right) \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function, x_t and h_{t-1} are the input and previous hidden state, and ω_f and b_f denote weight and bias parameters.

The activation output of Eq. (2) is bounded within the interval [0,1], with boundary values serving as binary gating signals. A null output (0) induces complete suppression of the preceding information, whereas unit output (1) facilitates perfect propagation through the temporal pathway. Subsequent to this gating operation, the system computes the state modification terms through two parallel transformations: (i) a sigmoidal regulatory layer (denoted as the input gate) that performs multiplicative modulation of the input stream, and (ii) a hyperbolic tangent transformation layer that generates a complementary candidate state vector. These components collectively implement the adaptive state update mechanism characteristic of LSTM architectures.

The input gate activation and the candidate value generation are defined as Eq. (3) and Eq. (4):

$$i_t = \sigma(\omega_i[h_{t-1}, x_t] + b_i) \tag{3}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{4}$$

where i_t denotes the activation of the input gate and $\tilde{c}_t$ refers to the vector of the new candidate values. By integrating these components, the updated cell state c_t is derived through the combination of Eq. (3) and Eq. (4), as expressed in Eq. (5):

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{5}$$

The final step involves generating the output. A sigmoid activation is first applied to determine which components of the cell state should influence the hidden state. The updated cell state is then transformed using a hyperbolic tangent function to constrain its values within the range $[-1, 1]$, and this result is scaled by the output gate's activation to produce the final hidden state.

Eq. (6) and Eq. (7) formalize this process, where Eq. (6) computes the output gate activation, and Eq. (7) derives the updated hidden state based on the modulated cell state.

$$o_t = \sigma(\omega_o[h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(c_t) \tag{7}$$

where o_t denotes the output gate activation obtained from the sigmoid function, and h_t represents the resulting hidden state. This hidden state is subsequently passed to the next time step or network layer, enabling the model to maintain temporal context across sequences.

3.3. Proposed Time2Vec-Enhanced LSTM Model

To effectively capture complex temporal dependencies in mobile traffic data, we integrate the Time2Vec mechanism with an LSTM architecture as illustrated in Fig. 2.

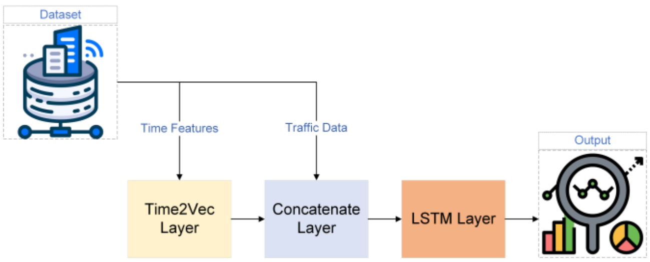

Fig. 2: The proposed hybrid Time2Vec-enhanced LSTM model

Traditional time encoding methods, such as one-hot or cyclical features, often fail to capture subtle periodic patterns, particularly when dealing with multiscale temporal trends. Time2Vec, a trainable time encoding function, addresses this limitation by learning both linear and periodic components of time explicitly, thereby improving the temporal awareness of the model.

3.3.1. Temporal Feature Extraction

In our architecture, raw temporal features such as hour of the day and day of the month, are first extracted from the timestamp. These temporal features are represented as a vector:

$$T = [Hour(t), Day(t)] \in \mathbb{R}^{d_t} \tag{8}$$

where T is a vector containing the extracted temporal features for each time step t, and d_t represents the dimensionality of the temporal feature vector.

These raw temporal features are then passed through a custom Time2Vec layer, which transforms them into a continuous vector representation. The Time2Vec layer outputs a concatenation of a linear transformation and several sine activations, allowing the model to capture both long-term linear trends and periodic cycles.

At each time step τ, the Time2Vec layer produces an embedding by combining a linear component and multiple periodic components as Eq. (9) below:

$$Time2Vec(T_\tau) = \begin{cases} \omega_0 \cdot T_\tau + b_0 & \text{if } i = 0 \\ \sin(\omega_i \cdot T_\tau + b_i) & \text{for } 1 \le i \le k \end{cases} \tag{9}$$

where ω_0 and b_0 are learnable scalar weights for the linear term, ω_i and $b_i \in \mathbb{R}^{d_t}$ are learnable parameters for the i^{th} periodic function, with $1 \le i \le k$, and T_τ is the temporal feature vector at time step τ. This allows the model to learn both short-term cycles (e.g., hourly and daily) and long-term trends (e.g., variations over multiple days).

3.3.2. Feature Normalization

To ensure that all input features contribute proportionally to model training, we apply z-score normalization to both the traffic volume and temporal features before sequence modeling. This standardization process transforms each feature by subtracting the mean and dividing by the standard deviation, as defined in Eq. (10). It centers the data around zero with unit variance, which helps stabilize training and accelerates convergence of the LSTM model.

$$z = \frac{x - \mu}{\sigma} \tag{10}$$

where x is the raw feature value, μ is the mean, and σ is the standard deviation of the feature.

3.3.3. Model Input Structure

Let the normalized historical traffic features at time step τ be represented as Eq. (11) below:

$$x_\tau = [u_\tau, \Delta_\tau, \Sigma_\tau] \in \mathbb{R}^3 \tag{11}$$

where u_τ, Δ_τ, and Σ_τ denote the uplink, downlink, and total traffic volumes, respectively. Let $T_\tau \in \mathbb{R}^3$ denote the temporal feature vector at time step τ, which includes hour and day.

To incorporate recurring temporal patterns, we apply the

Time2Vec embedding $\phi(T_\tau) \in \mathbb{R}^{k+1}$ as defined in Eq. (9). We then concatenate the traffic vector x_τ with the temporal embedding to form the combined input defined as Eq. (12) below:

$$z_\tau = [x_\tau \parallel \phi(T_\tau)] \in \mathbb{R}^{3+k+1} \tag{12}$$

Over a sliding window of length L, we construct the input sequence as Eq. (13):

$$Z = [z_{\tau-L+1}, \dots, z_\tau] \in \mathbb{R}^{3+k+1} \tag{13}$$

This sequence is then passed into an LSTM layer, which models temporal dependencies in both traffic and time-aware patterns and is defined as Eq. (14):

$$h = LSTM(Z) \tag{14}$$

The LSTM output h is finally used to predict the traffic volumes at the next time step through a fully connected layer.

3.3.4. Output and Training Strategy

The final hidden state output from the LSTM, denoted as $h \in \mathbb{R}^d$, is passed through a fully connected (dense) layer to generate the predicted traffic volumes at the next time step:

$$\hat{y}_{\tau+1} = W_{out} \cdot h + b_{out} \tag{15}$$

where $W_{out} \in \mathbb{R}^{3\times d}$ is the learnable weight matrix projecting the LSTM hidden state $h \in \mathbb{R}^d$ to the output space, and $b_{out} \in \mathbb{R}^3$ is the learnable bias vector. The resulting prediction $\hat{y}_{\tau+1} = [\hat{u}_\tau, \hat{\Delta}_\tau, \hat{\Sigma}_\tau]$ corresponds to the uplink, downlink, and total traffic volumes at the next time step.

4. Dataset and Experimental Setup

4.1. Dataset Description

The dataset used in this study was collected from a telecom service provider operating in Kandahar Province, Afghanistan, and contains traffic data for an LTE base station with three cells. The data spans the entire month of November 2024, from November 1 to November 30, with continuous recording for 24 hours each day.

The dataset includes parameters such as Date and Time, Cell ID (which specifies one of the three cells), Uplink Traffic Volume (a measurement of data uploaded by users within a cell), Downlink Traffic Volume (a measurement of data downloaded by users within a cell), and Total Traffic Volume (the sum of uplink and downlink traffic, representing overall data consumption within each cell).

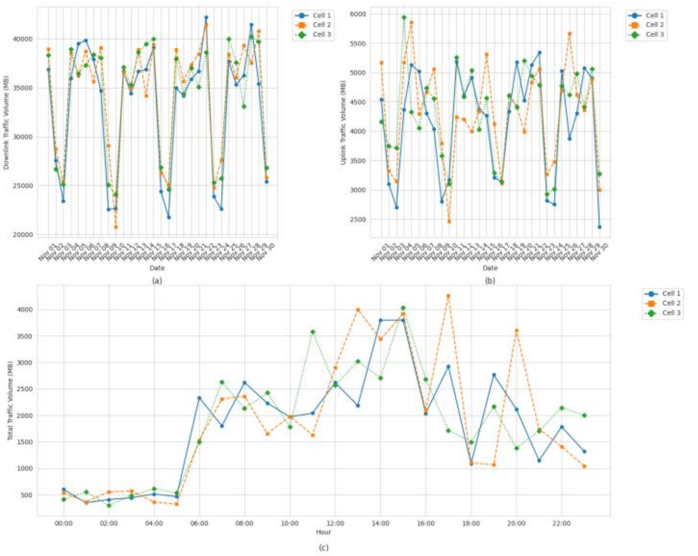

Fig. 3: Traffic volume analysis: (a) Daily Download Traffic Volume per cell, (b) Daily Upload Traffic Volume per cell, and (c) Hourly Total Traffic Volume per cell (November 15, 2024).

The dataset comprises a total of 2,160 samples, corresponding to hourly measurements collected across all three cells over the 30-day period. Different views of the dataset are shown in Fig. 3, and a summary of the descriptive statistics for these traffic parameters is provided in Table 1.

Table 1. Summary statistics of the dataset

Traffic Type	Minimum (MB)	Maximum (MB)	Mean (MB)	Standard Deviation (MB)
Uplink	11.39	800.50	177.09	132.81
Downlink	153.84	5,120.32	1,408.95	937.84
Total	173.9	5,920.82	1,586.04	1,037.09

4.2. Data Preprocessing

4.2.1. Train-Test Split

The dataset was split randomly into training and testing portions to enable the validation of the actual performance of the model and its ability to generalize over unseen data, as shown in Table 2. Special consideration was made to set aside 20% of the records as a test set, so that the model could be trained with 80% of the data. This kind of division is employed in real applications where a model is trained on historical data, then deployed to predict on future or unseen data. This random partition confines the elements present in both partitions to similar distributions and hence opposes the possibility of overfitting, giving a strong validation of the model.

Table 2. Train-test split of total traffic data

Feature	Total Samples	Training Set (80%)	Testing Set (20%)
Uplink Traffic	2160	1728	432
Downlink Traffic	2160	1728	432
Total Traffic	2160	1728	432

4.2.2. Feature Selection

From the raw dataset, we select two groups of features for modeling:

- Traffic-related features: uplink traffic volume, downlink traffic volume, and total traffic volume per hour.
- Temporal context features: hour of the day and day of the month, derived from the timestamp.

Traffic characteristics indicate the system's load level at each time, whereas temporal features provide the contextual setting for recurring patterns. The temporal features are fed into a Time2Vec embedding layer (discussed in Section 3) to encode linear as well as periodic time trends.

To ensure that all features contribute proportionally to the learning process, they are normalized using z-score standardization, as discussed in Section 3.3.2. We chose z-score over alternatives like min-max [20], as it better preserves variability and handles outliers, providing more stable inputs for the proposed model.

4.3. Evaluation Metrics

To evaluate the performance of the proposed Time2Vec-enhanced LSTM model for cellular traffic prediction, two commonly used evaluation metrics, namely RMSE and R^2, are employed. RMSE measures the average magnitude of the prediction error, while the R^2 describes exactly how much of the variance in the true data is explained by the model. Both predictive accuracy and explanatory power are incorporated into our evaluation using these metrics.

4.3.1. Root Mean Square Error (RMSE)

The RMSE represents the square root of the mean of squared deviation between forecasted traffic volume and actual observed traffic volume as described in Eq. (16). It is expressed with the same units as those of the observed traffic values (MB). So, the lower the RMSE values, the higher are the accuracies of the predicted values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{16}$$

where y represents actual target value, $\hat{y}$ denotes the predicted value, and n is the total number of observations.

4.3.2. Coefficient of Determination (R^2)

R^2 evaluates how well the variations existing in the actual traffic data are explained by the model. R^2 does possess values close to 1 when the model is able to capture the temporal patterns well, and the values approach 0 when the explanatory power is limited. R^2 is expressed in Eq. (17) as shown below:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{17}$$

where $\bar{y}$ represents the mean of actual target values.

5. Results and Discussion

To evaluate the effectiveness of the proposed Time2Vec-enhanced LSTM model, experiments were conducted on a dataset comprising hourly cellular traffic data collected over a 30-day period as discussed in Section 4.1. The model was designed to forecast three target variables: uplink, downlink, and total traffic volumes, using both historical traffic patterns and temporal contextual features.

The Time2Vec layer was applied to the two available time-related features, hour and day, before concatenation with the traffic volume sequences. This temporal embedding captures both linear and periodic dependencies, which are crucial for modeling daily patterns and hourly fluctuations commonly observed in network traffic data.

5.1. Overall Performance Comparison

The predicted and actual traffic values for all three targets are visually compared in Fig. 4. The plots indicate that the predicted values closely follow the overall trends and periodic fluctuations of the true values, particularly for total traffic, where the model captures both sharp peaks and troughs effectively. Although some discrepancies are visible, the alignment between actual and predicted patterns supports the suitability of Time2Vec for encoding periodic temporal signals.

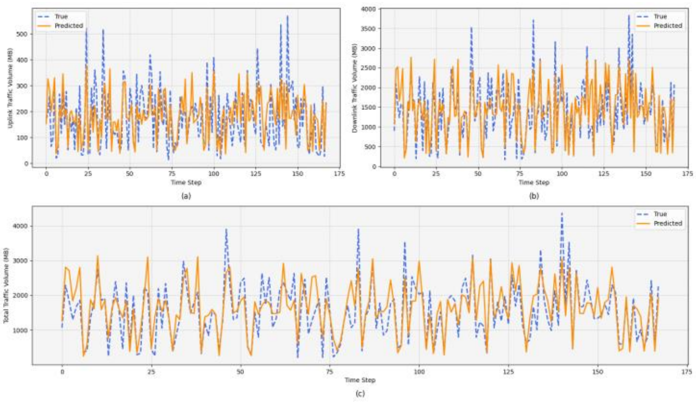

Fig. 4: Comparison of actual and predicted traffic volumes for (a) Uplink, (b) Downlink, and (c) Total Traffic over 175 time steps

Among the considered methods, the DL approaches (LSTM and GRU) demonstrated better performance than the traditional statistical and ML models, ARIMA and SVR, respectively, particularly in terms of R^2 scores for downlink and total traffic. However, both LSTM and GRU still fell short of the performance achieved by the proposed model, which achieved the highest R^2 across all traffic types.

Table 3. Performance comparison of different models

Model	Traffic Type	RMSE (MB)	R^2 Score
Time2Vec-enhanced LSTM	Uplink	73.48	0.5991
	Downlink	468.52	0.6254
	Total	498.53	0.6531
LSTM	Uplink	92.25	0.3686
	Downlink	501.35	0.5711
	Total	539.62	0.5936
GRU	Uplink	94.78	0.3335
	Downlink	524.96	0.5298
	Total	547.22	0.5820
ARIMA	Uplink	103.10	0.3800
	Downlink	576.54	0.5333
	Total	604.65	0.5820
SVR	Uplink	108.81	0.1216
	Downlink	737.66	0.0308
	Total	828.70	0.0415
RF	Uplink	95.53	0.3228
	Downlink	533.93	0.5136
	Total	541.50	0.5907

The performance of the proposed Time2Vec-enahanced LSTM model was evaluated using RMSE and R^2 metrics and benchmarked against a diverse set of models, including LSTM, GRU, ARIMA, SVR, and RF. As illustrated in Table 3, the proposed model showed the best prediction accuracy with the least RMSE and maximum R^2, for all traffic types (uplink, downlink, and total), thus convincingly speaking for a better capturing of the temporal dynamics from cellular traffic data.

While ARIMA showed slightly better R^2 than GRU in the uplink case, its overall RMSE values remained higher. SVR performed the poorest across all metrics, with the highest RMSE and the lowest R^2 scores. The RF, meanwhile, produced mixed results. Its RMSE values were higher than those of LSTM and only marginally better than GRU for total traffic. Its R^2 scores generally lagged behind LSTM and GRU, however, for total traffic, its R^2 was slightly better than that of GRU. Despite having better performance comparing to ARIMA and SVR, it still showed slightly worse R^2 than ARIMA in the uplink and downlink cases. This indicates that although RF sometimes reduced average prediction error slightly compared to the weaker models, it struggled to capture the temporal patterns and variability in the data, often regressing toward the mean and failing under conditions of high fluctuation.

This set of results justifies the integration of learning temporal embeddings through Time2Vec into LSTM for further spatiotemporal modeling of cellular traffic. To provide a clearer comparison of performance, Fig. 5 visualizes the R^2 scores across all models for each traffic type, highlighting the consistent superiority of the proposed Time2Vec-enhanced LSTM model. It presents a grouped bar chart of R^2 values, with separate bars for uplink, downlink, and total traffic types across all evaluated models.

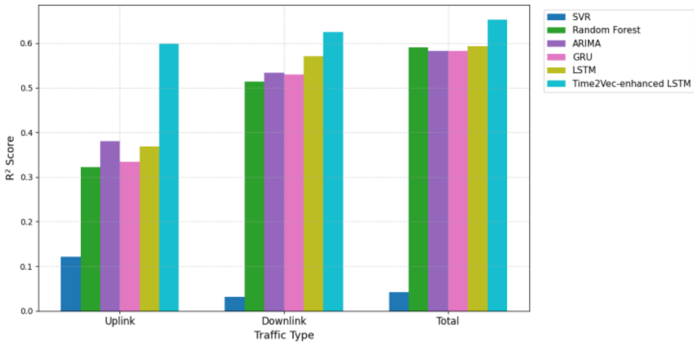

Fig. 5: R^2 scores of all considered models across traffic types

5.2. Ablation Study on Temporal Encoding

To better understand the impact of temporal feature encoding on model performance, an ablation study was conducted by comparing four LSTM-based architectures:

1) LSTM model using only traffic data (no time features),
2) LSTM with raw time inputs (hour and day),
3) LSTM with fixed cyclical encodings (sin/cos transformations),
4) the proposed Time2Vec-enhanced LSTM.

The performance differences among these models are illustrated in Fig. 6.

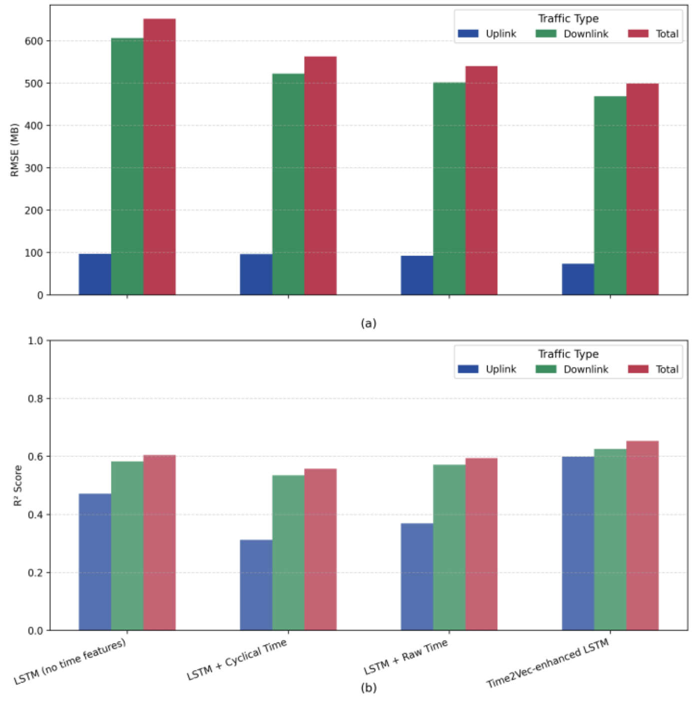

Fig. 6: Ablation study comparing (a) RMSE and (b) R^2 score across four LSTM-based models for uplink, downlink, and total traffic prediction

As shown in Fig. 6 (a) and (b), the inclusion and design of time features significantly affect the predictive performance. The LSTM with no time feature was found to have only moderate predictive capacity and is particularly limited in its ability to capture temporal dependencies inherent in the data. The inclusion of cyclical time encodings resulted in slightly reduced RMSE values for downlink and total traffic in comparison to the LSTM with no time feature, while it tended to have reduced R^2 scores for all traffic types. This would indicate that fixed periodic encodings do not properly capture the temporal variability in the data. On the other hand, LSTM with the raw hour and day features was better in some cases than the basic LSTM and its cyclical variants, indicating that whatever the model can glean from raw temporal inputs is worthwhile.

In contrast, the Time2Vec-enhanced LSTM consistently achieved the lowest RMSE and highest R^2 scores across all traffic types, clearly outperforming all other variants. These results confirm that learnable temporal embeddings from Time2Vec provide a more expressive and adaptive representation of time, enabling the model to better capture both short-term patterns and long-term periodic trends in the traffic data.

6. Conclusions

In this study, a hybrid Time2Vec-enahanced LSTM model was proposed for traffic volume prediction in cellular networks. The model simultaneously integrates periodic temporal features with historical traffic data so that independences of short-term changes and long-term cyclic patterns can be learned.

Experiments indicated that the proposed Time2Vec-enhanced LSTM model is dominant compared to considered DL, time series, and ML methods for uplink, downlink, and total traffic volume prediction. The model achieved lower RMSE and higher R^2 scores across all traffic types, consistently outperforming all the models. Additionally, the ablation study confirmed that the Time2Vec-based temporal encoding was the key factor driving this superior performance, as models without time features or using only raw or cyclical time inputs consistently showed higher errors and lower R^2 scores. Its strong performance is attributed to the inclusion of learnable periodic transformations via Time2Vec, which significantly enhanced the LSTM's capacity to recognize cyclical patterns in the data.

Summing up, the findings corroborate the importance of temporal embeddings in DL-based traffic prediction. The proposed Time2Vec-enhanced LSTM architecture appears to be a potential candidate for intelligent network traffic prediction.

In the future, research may be undertaken to extend the model to multi-cell scenarios and to experiment with more time features such as weekends, holidays, and special events, while hyperparameter tuning could be applied to achieve improved prediction accuracy and performance.

References

[1] J. Xiao, Y. Cong, W. Zhang, and W. Weng, "A cellular traffic prediction method based on diffusion convolutional GRU and multi-head attention mechanism," *Cluster Computing*, vol. 28, no. 2, p. 125, 2025.

[2] H. Riaz, S. Öztürk, and A. Çalhan, "A Robust Handover Optimization Based on Velocity-Aware Fuzzy Logic in 5G Ultra-Dense Small Cell HetNets," *Electronics*, vol. 13, no. 17, Art. no. 17, Jan. 2024, doi: 10.3390/electronics13173349.

[3] H. Riaz, S. Öztürk, S. Aldirmaz-Colak, and A. Çalhan, "A Handover Decision Optimization Method Based on Data-Driven MLP in 5G Ultra-Dense Small Cell HetNets," *J Netw Syst Manage*, vol. 33, no. 2, p. 31, Feb. 2025, doi: 10.1007/s10922-025-09903-6.

[4] O. Aouedi, V. A. Le, K. Piamrat, and Y. Ji, "Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions," *ACM Comput. Surv.*, vol. 57, no. 6, p. 151:1-151:37, Feb. 2025, doi: 10.1145/3703447.

[5] X. Wang *et al.*, "A Survey on Deep Learning for Cellular Traffic Prediction," *Intelligent Computing*, vol. 3, p. 0054, Jan. 2024, doi: 10.34133/icomputing.0054.

[6] H. Riaz, P. Güneş, H. Benli, and F. H. Ahmadzai, "A Comparative Analysis of Machine Learning Models for Cellular Load Prediction: Insights from Real-World Data," *Journal of Emerging Trends in Engineering Research*, vol. 13, no. 4, pp. 73–81, 2025.

[7] A. Azari, P. Papapetrou, S. Denic, and G. Peters, "Cellular traffic prediction and classification: A comparative evaluation of LSTM and ARIMA," presented at the Discovery Science: 22nd International Conference, DS 2019, Split, Croatia, October 28–30, 2019, Proceedings 22, Springer, 2019, pp. 129–144.

[8] S. Medhn, B. Seifu, A. Salem, and D. Hailemariam, "Mobile data traffic forecasting in UMTS networks based on SARIMA model: The case of Addis Ababa, Ethiopia," presented at the 2017 IEEE AFRICON, IEEE, 2017, pp. 285–290.

[9] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Interactive Temporal Recurrent Convolution Network for Traffic Prediction in Data Centers," *IEEE Access*, vol. 6, pp. 5276–5289, 2018, doi: 10.1109/ACCESS.2017.2787696.

[10] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019, doi: 10.1109/JSAC.2019.2904363.

[11] C. Gijón, M. Toril, S. Luna-Ramírez, M. L. Marí-Altozano, and J. M. Ruiz-Avilés, "Long-Term Data Traffic Forecasting for Network Dimensioning in LTE with Short Time Series," *Electronics*, vol. 10, no. 10, Art. no. 10, Jan. 2021, doi: 10.3390/electronics10101151.

[12] S. Jaffry and S. F. Hasan, “Cellular traffic prediction using recurrent neural networks,” presented at the 2020 IEEE 5th international symposium on telecommunication technologies (ISTT), IEEE, 2020, pp. 94–98.

[13] H. Chekireb, L. Fergani, S. A. Selouani, M. R. Deramchi, and R. Rochedi, “Improving LTE network Retainability KPI prediction performance using LSTM and Data Filtering technique,” presented at the 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA), IEEE, 2022, pp. 1–6.

[14] W. Jiang, M. He, and W. Gu, “Internet Traffic Prediction with Distributed Multi-Agent Learning,” *Applied System Innovation*, vol. 5, no. 6, Art. no. 6, Dec. 2022, doi: 10.3390/asi5060121.

[15] T. Xu and Y. Yan, “Cell base station traffic prediction based on GRU,” *Comput. Perform. Commun. Syst*, vol. 7, no. 1, pp. 66–72, 2023.

[16] K. Gao *et al.*, “Predicting Traffic Demand Matrix by Considering Inter-flow Correlations,” in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 165–170. doi: 10.1109/INFOCOMWKSHPS50562.2020.9163001.

[17] D. Aloraifan, I. Ahmad, and E. Alrashed, “Deep learning based network traffic matrix prediction,” *International Journal of Intelligent Networks*, vol. 2, pp. 46–56, Jan. 2021, doi: 10.1016/j.ijin.2021.06.002.

[18] S. M. Kazemi *et al.*, “Time2vec: Learning a vector representation of time,” *arXiv preprint arXiv:1907.05321*, 2019.

[19] K. Zhou, C. Zhang, B. Xu, J. Huang, C. Li, and Y. Pei, “TE-LSTM: A Prediction Model for Temperature Based on Multivariate Time Series Data,” *Remote Sensing*, vol. 16, no. 19, Art. no. 19, Jan. 2024, doi: 10.3390/rs16193666.

[20] H. Riaz, S. Öztürk, S. A. Çolak, and A. Çalhan, “Performance Analysis of Weighting Methods for Handover Decision in HetNets,” *Gazi University Journal of Science*, vol. 37, no. 4, pp. 1791–1810, Jan. 2024, doi: 10.35378/gujs.1373452.