Journal of Applied Data Sciences
Vol. 6, No. 1, January 2025, pp. 20-33

Cellular Traffic Prediction Models Using Convolutional Long Short-Term Memory

A Sunil Samson[1,*], N Sumathi[2], Siti Sarah Maidin[3], Qingxue Yang[4]

[1,2] *IT Department, Sri Ramakrishna College of Arts and Science, Coimbatore, India*

[3]*Faculty of Data Science and Information Technology (FDSIT), INTI International University, Nilai, Malaysia*

[3]*Department of IT and Methodology, Wekerle Sandor Uzleti Foiskola Budapest, Hungary*

[4]*Faculty of Liberal Arts, Shinawatra University, Thailand*

(Received: September 11, 2024; Revised: October 31, 2024; Accepted: November 8, 2024; Available online: December 27, 2024)

Abstract

Precise cellular traffic modeling and prediction is essential to future big data-based cellular network management for providing autonomic control and user-satisfied stable mobile services. However, the traditional methods have difficulty learning the complex hidden patterns of the users' traffic data from cross-domains because of their shallow learning characteristics. Deep learning (DL)-based methods could somewhat identify these hidden patterns by learning the underlying spatial and temporal features and their dependencies. Yet, they too have constraints in handling the noisy and sparse data, reducing the prediction accuracy with increased computation time and associated storage costs. Therefore, this paper presents an intelligent cellular traffic prediction model (ICTPM) using two improved deep learning algorithms to tackle the negative impacts of noisy and sparse traffic datasets. Firstly, the Enhanced Stacked Denoising Auto-Encoder (ESDAE) is introduced to eliminate the noise in the traffic data by an adaptive Morlet wavelet transform. Secondly, Multi-dimensional Spatiotemporal Sparse-representation Convolutional Long Short-Term Memory (MDSTS-CLSTM) is used to learn the hidden patterns by extracting the spatial-temporal dependencies and predict the cellular usage in the presence of data sparsity problem. This MDSTS-CLSTM is developed by combining the Long Short-Term Memory (LSTM) with the Convolutional Neural Networks (CNN) and improvising the multi-dimensional feature learning, spatial-temporal analysis, and sparse representation properties of the hybrid DL algorithm. Evaluated over real-world cellular traffic cross-domain datasets from Telecom Italia and Open-CellID, the proposed ICTPM outperforms the state-of-the-art methods with 5-10% better performance enhancements.

Keywords: Cellular Traffic Prediction, Cross-Domain Big Data, Enhanced Stacked Denoising Auto-Encoder, adaptive Morlet wavelet transform, Convolutional Long Short-Term Memory, Multi-dimensional Spatiotemporal Sparse-representation learning, Process Innovation, Product Innovation

1. Introduction

The growth of telecommunication technologies enhanced the speed of network connectivity and increased mobile users. The extensive growth of mobile operators and internet usage resulted in high network traffic. As per the Ericsson report, the mobile network traffic will exceed 104.4 Exabyte monthly in 2025. Hence, cellular traffic must be predicted to provide better service without delay. The Cellular Traffic Prediction (CTP) model analyses the network flow to provide better service for mobile users [1]. The traffic prediction model enables the network operators to manage the network resources and make appropriate decisions for effective utilization of resources, congestion control and bandwidth allocation. The traffic patterns are dynamic, and it is hard to analyze the network traffic. In addition, determining of correlation between the spatial and temporal properties to predict network traffic adds more complexity. Several works have been developed earlier for predicting cellular traffic using Machine learning (ML) and deep learning (DL) methods [2]. Many ML algorithms, like support vector machine (SVM), artificial neural network (ANN), auto-regressive integrated moving average (ARIMA), etc., have been developed in recent years to analyze traffic datasets. However, there are some limitations in the prevailing approaches while analyzing the network traffic dataset. The inappropriate selection of ML methods causes the limitations, datasets with huge noise and usage of unsuitable

***Corresponding author: A Sunil Samson (sunilsamson@irins.org)**
DOI: https://doi.org/10.47738/jads.v6i1.472

feature selection methods [3]. This leads to higher processing time, reduced traffic prediction accuracy and a high error rate. Moreover, the shallow learning characteristics of the ML model do not learn the hidden features of traffic data, leading to a high error rate. To overcome these issues, DL methods are used for traffic prediction.

DL algorithms such as convolutional neural networks (CNN), long short-term memory (LSTM), recurrent neural networks (RNN), etc., are conventionally used for CTP [4]. This model learns the hidden features from cross-domain effectively. The DL methods are more beneficial for learning the spatial-temporal features and their dependencies. This feature increases traffic prediction accuracy. Furthermore, DL models can handle large datasets, but when the dataset incorporates noise and sparse representations, the prediction accuracy of the DL models becomes unreliable. Therefore, the noise and data sparsity problem must be resolved to make the DL models more reliable. This paper addressed the limitations of CTP and proposed ICTPM based on ESDAE and MDSTS-CLSTM methodologies. Initially, the noise in the network traffic dataset is removed utilizing ESDAE. Then, the cellular traffic is predicted effectively by processing the denoised data using the MDSTS-CLSTM model with its superior properties like sparse representation, multi-dimensional learning and spatial-temporal analysis. The main objectives of this paper are to remove the noise in traffic data using the ESDAE method and include multiple properties such as multi-dimensional, sparse representation and spatial-temporal analysis in the CLSTM classifier to extract the features and predict cellular traffic data. The rest of the article is structured as follows: literature study in section 2. Proposed feature selection and classification approach with its implementations in section 3. Performance metrics are evaluated in section 4. Results and conclusion in section

2. Related Work

CTP has recently become one of the most vital mobile communication technologies research topics. ML and DL algorithms have been predominantly utilized for CTP due to their extensive feature learning and processing capabilities. Zhang et al. [5] proposed a Spatial-Temporal Cross-domain Neural Network (STCNet) to predict mobile traffic. The STCNet model is constructed by combining CNN and LSTM models, which capture the spatial and temporal features to predict cellular traffic. This model is evaluated using a real-time cellular traffic dataset and attained a performance improvement of 4 to 13%. However, this model has a complex structure. Lin et al. [6] proposed a multi-graph convolutional network and LSTM (MGCN-LSTM) technique for CTP. Here, the attention mechanism is applied to the LSTM model to enhance the traffic prediction accuracy. Also, the capacity of macro base stations is determined using the clustering approach and Multilayer Perceptron (MLP) method to reduce the power consumption. This model is examined utilizing the Milan city network traffic dataset, and MGCN-LSTM gained better prediction performance but has a complex structure as a drawback. By integrating GCN and GRU models, Zhang et al. [7] suggested a spatial-temporal graph convolutional gated recurrent unit (STGCGRU) model for CTP. This method is evaluated on the GEANT dataset and achieved an accuracy of 91%, MAE of 0.00279, RMSE of 0.0069 and R-square of 0.88 for the execution time of 15 minutes. However, the accuracy of this model is highly dependent on training time. Shen et al. [8] employed CNN methodology for CTP with a time-wise attention mechanism applied to the CNN model to extract the temporal and spatial features. This model is tested on the Milan dataset and gained improved training efficiency with an average training time of 144.57s, but it has also increased computational cost. Yao et al. [9] proposed a Multi-View Spatial-Temporal Graph Network (MVSTGN) for CTP by combining convolutional and attention mechanisms for analyzing traffic patterns. This combination extracts the spatial and temporal attributes more effectively, attaining higher prediction accuracy with slightly high computational complexity. Zeng et al. [10] developed an attention-based multi-component spatial-temporal cross-domain neural network (A-MCSTCNet) for CTP using either Conv-LSTM structure or Conv-GRU structure to predict the traffic data. On experimenting, A-MCSTCNet with Conv-GRU structure performed better for traffic prediction and gained the performance improvement of 38.79 to 103.17% for the internet dataset, 12.24 to 24.89% for call and 14.56 to 55.82 for SMS for the Milan dataset. However, this model takes longer training time.

Balamurugan et al. [11] implemented an Enhanced Deep Reinforcement Learning (EDRL) methodology constructed using MLP and Monte Carlo learning (MCL) approach for CTP. The performance of this model is assessed using encrypted and non-encrypted network traffic datasets and achieves a higher accuracy of 97.20%, but this model requires more data for training. Duan et al. [12] analyzed CTP using a Generative Adversarial Network (GAN) framework called CrowdGAN. It uses the Conv-LSTM model to extract the spatial and temporal features. The effectiveness of this model is analyzed using two real-time traffic datasets, and noted that it had reduced the RMSE value by 47%. However, this model is complex to train. Fang et al. [13] constructed a Spatiotemporal Graph Neural Network (SDGNet) by integrating dynamic GCN and gated linear unit (GLU) models to extract the spatial and temporal features ofCTP. This model is evaluated on Turkcell real world dataset and obtained the MAE value of 0.78 and RMSE value of 1.28 with the time duration of 30 min. However, accuracy of this model is limited with the training period. Gao [14] utilized the Smoothed LSTM (SLSTM) model for 5G network CTP by temporal feature extraction and analysis by auto-correlation method. The dataset is collected from foreign network operators and achieved RMSE of 3014.2, MAE of 287.3 and R-squared of 0.83, respectively. However, this model does not consider the factors affecting the 5G network during traffic prediction. Lin and Nuha [15] improved CTP accuracy using One-dimensional CNN (1DCNN) and GRU. This model is examined on the Kaggle dataset of Italian network traffic and attained the RMSE of 0.024, MAE of 0.021 and MAPE of 12.10% for predicting the traffic in 3350 cells. However, the effectiveness of this model is not evaluated for large datasets.

Selvi and Thamilselvan [16] deployed the GRU method integrated with diffusion convolution operation to extract the spatial and temporal features for CTP. The stochastic gradient-related scheduled sampling approach enhances the prediction model's performance with an optimal decay rate. This method improved accuracy from 87% to 94% with a reduced error rate of 7.98% while examining the network traffic dataset. However, this model has a low convergence and learning rate. Shawel et al. [17] developed a CTP model using a multivariate hybrid CNN-LSTM methodology. This method achieved the RMSE value of 0.81 and MAPE of 2.97 in the network traffic dataset, which is lower than the prevailing traffic prediction models. However, this model has a complex structure. Zhou [18] developed a CTP model using GCN for learning spatial features, and the GCN integrated with an attention mechanism to extract the temporal features. This model is experimented with two real-time datasets and noted that it uses fewer epochs during training. Chen et al. [19] presented STP-GLN, a spatial-temporal parallel prediction model based on Graph CNN (GCNN) and LSTM Networks. Evaluated on cellular network traffic datasets, this model improved RMSE by 81.7%, the MAE by 82.7%, and the R-squared (R2) by 2.2%. Nie et al. [20] proposed a Reinforcement Learning (RL)-based CTP approach that combines Deep Q-learning (DQN) and GAN for feature extraction. This model achieved 83% performance improvement when evaluated on three datasets.

The methods illustrated in the literature have shown that the major limitation that needs to be tackled is the model complexity and the requirement for more training data. Therefore, the noise and data sparsity problem must be handled effectively along with the model complexity. This paper develops ICTPM using ESDAE for denoising and MDSTS-CLSTM for sparsity-aware traffic prediction.

3. Methodology

The proposed cellular prediction model is the integration of ESDAE and MDSTS-CLSTM methodologies. In the first stage, the network traffic dataset is pre-processed using the ESDAE method that uses adaptive Morlet wavelet transform to remove the noise in the dataset. In the second stage, the noise-removed data is deployed for learning the hidden patterns of the dataset utilizing the MDSTS-CLSTM method. It is created by fusing the properties of CNN and LSTM models. This method uses the benefits of different properties such as multi-dimensional, sparse representation and spatial-temporal analysis. These properties extract the most relevant features from the cross-domain traffic datasets and enhance the performance of the traffic prediction model. The cellular traffic and cross-domain datasets are given to the ESDAE block to remove noise. Further, CLSTM blocks with features like multi-dimensional learning, sparse representation and spatial and temporal analysis are used to learn the spatial and temporal features for predicting the cellular traffic data. The framework of the proposed cellular prediction model is displayed in Figure 1.

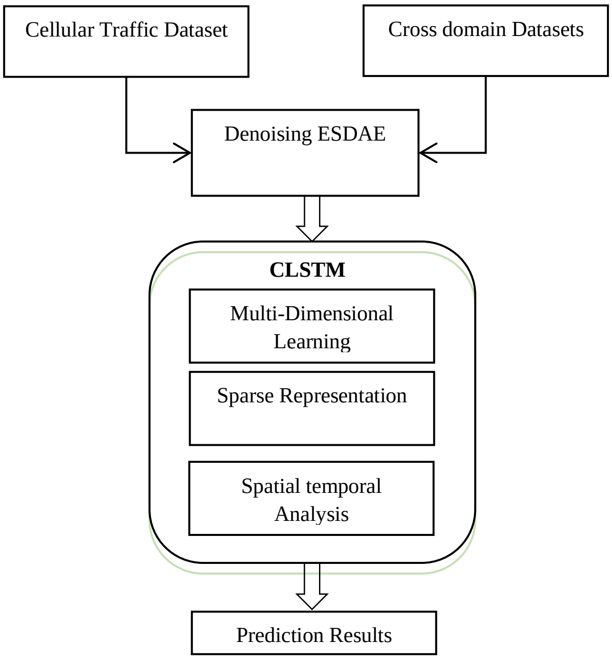

Figure 1. Framework of Proposed ICTPM

3.1. Dataset Description

3.1.1. Milan City Traffic Data

The performance of the proposed model is evaluated utilizing the cellular traffic dataset of European telephone service provider Telecom Italia. The traffic data was gathered in Milan from November 10, 2013, to January 1, 2014. The data is collected for 10 min. This dataset contains 10,000 base stations. From this, 900 BS is taken as input for testing the model. The dataset is divided into 90% for training and 10% for testing. The used spatial-temporal CDR database incorporates the data of Grid ID, time stamp, internet activity, outbound call activity, inbound call activity, and outbound and inbound SMS activity, which all come under three services, namely Internet service, call service and Short Message Service (SMS). The CDR database does not specify the network activity in terms of units. For instance, if more SMS is received or sent, the magnitude of SMS activity is high. Here, the area of Milan is partitioned as a grid overlay of height and width $(H \times W)100 \times 100$, and it occupies an area of 235×235 meters; it is referred to as cells. These cells store the records of the services, as mentioned earlier. The particular service type is represented as $s \in \{SMS, Call, Internet\}$, and the cellular traffic is the representation of a spatiotemporal sequence of data points $D_s = \{D_{s,t} | t = 1,2,3, \ldots .T\}$, here T represents the total number of time intervals. $D_{s,t}$ specifies the traffic matrix at the time interval t of geographical area$(H \times W)$, and it is expressed as,

$$D_{s,t} = \begin{bmatrix} d_{s,t}^{(1,1)} & d_{s,t}^{(1,2)} \ldots & d_{s,t}^{(1,W)} \\ d_{s,t}^{(2,1)} & d_{s,t}^{(2,2)} \ldots & d_{s,t}^{(2,W)} \\ d_{s,t}^{(H,1)} & d_{s,t}^{(H,2)} \ldots & d_{s,t}^{(H,W)} \end{bmatrix} \tag{1}$$

Here, $d_{s,t}^{(H,W)}$ used for measuring the volume of cellular traffic in a cell with coordinates (h, w) and the sequence is denoted as $D_s \in R^{T \times H \times W}$. The representation of service type is not included in further steps, and it is specified as $d_{s,t}^{(h,w)} = d_t^{(h,w)}$ and $D_{s,t} = D_t$ for ease of readability. The spatial and temporal dynamics and correlation analysis of traffic datasets are explained as follows. The autocorrelation of SMS in a particular cell h, w is calculated as,

$$\mathrm{r_k} = \frac{\sum_{t=1}^{T-k}\left(\left(d_t^{(h,w)} - \bar{d}^{(h,w)}\right)\left(d_{t+k}^{(h,w)} - \bar{d}^{(h,w)}\right)\right)}{\sum_{t=1}^{T}(d_t^{(h,w)} - \bar{d}^{(h,w)})^2} \quad ,0 \leq k \leq T \tag{2}$$

Where, $\bar{d}^{(h,w)}$denotes the average value cell in the time domain. The spatial correlations of traffic data are calculated using Pearson correlation coefficient (ρ), and it is expressed as,

$$\rho = \frac{\mathrm{cov}\,(d^{(h,w)}, d^{(h',w')})}{\sigma_{d^{(h,w)}}\sigma_{d^{(h',w')}}} \tag{3}$$

Here, $cov(.)$ indicates the covariance operator, σ represents the standard deviation.

3.1.2. Cross-domain Datasets

The volume of cellular traffic could be influenced by other factors like the count of Base Stations (BS) and point of Interconnectivity (POI) of cells rather than spatiotemporal factors. Therefore, different factors must be considered while predicting cellular traffic based on cross-domain datasets. In addition to the mentioned factors, the cell's social activities reflect the user's service utility, which is also used for predicting traffic. Hence, this work uses the data of POI, social activity level and BS to predict traffic data. The BS data is procured from Open-CellID, POI from Google Places API and social activity from Dandelion API.

The BS data includes location details, the coverage range estimated for each BS and the mobile country code. The information of BS is expressed as,

$$D_{BS} = \begin{bmatrix} d_{BS}^{(1,1)} & d_{BS}^{(1,2)} & \dots & d_{BS}^{(1,W)} \\ d_{BS}^{(2,1)} & d_{BS}^{(2,2)} & \dots & d_{BS}^{(2,W)} \\ \vdots & \vdots & & \vdots \\ d_{BS}^{(H,1)} & d_{BS}^{(H,2)} & \cdots & d_{BS}^{(H,W)} \end{bmatrix} \tag{4}$$

The POI data is collected from 13 locations like stores, subway stations, restaurants, etc. The final representation is formed by adding the number of each category together. The matrix form of the POI dataset is expressed below,

$$D_{POI} = \begin{bmatrix} d_{POI}^{(1,1)} & d_{POI}^{(1,2)} & \dots & d_{POI}^{(1,W)} \\ d_{POI}^{(2,1)} & d_{POI}^{(2,2)} & \dots & d_{POI}^{(2,W)} \\ \vdots & \vdots & & \vdots \\ d_{POI}^{(H,1)} & d_{POI}^{(H,2)} & \cdots & d_{POI}^{(H,W)} \end{bmatrix} \tag{5}$$

The degree of user demand is reflected in the social activity of a cell. The social activity level data includes user-generated data on Twitter, including keywords and locations. The data is pre-processed and represented in matrix form as,

$$d_{Social} = \begin{bmatrix} d_{Social}^{(1,1)} & d_{Social}^{(1,2)} & \dots & d_{Social}^{(1,W)} \\ d_{Social}^{(2,1)} & d_{Social}^{(2,2)} & \dots & d_{Social}^{(2,W)} \\ \vdots & \vdots & & \vdots \\ d_{Social}^{(H,1)} & d_{Social}^{(H,2)} & \cdots & d_{Social}^{(H,W)} \end{bmatrix} \tag{6}$$

Here, $d_{Social}^{(h,w)}$ denotes the count of social activity of a cell (h, w).

3.2. Pre-Processing

The noise in the network traffic dataset is denoised in the pre-processing step using the Enhanced Stacked Denoising Auto-Encoder (ESDAE). The proposed auto-encoder is constructed utilizing the Adaptive Morlet Wavelet Transform, which reconstructs the given input by modifying the model's parameters to enhance the traffic prediction accuracy. The standard AE has encoder and decoder parts. The encoder incorporates decreasing hidden layers, which uses weight and bias to encode the data, and the decoder attempts to reconstruct the actual data with the increasing hidden layers. When the AEs have more hidden layers than the input, they reflect the same input data at the output stage and do not provide any useful information. This limitation is overcome with a denoising autoencoder. This kind of AE randomly corrupts the input or includes noise, and this procedure forces the AE to reconstruct the actual input. Hence, the impact of noise in network traffic is effectively handled using a DAE. This type of AE uses multiple layers, requiring more parameters to tune during the training process. This creates computational complexity and increases the training time. This issue is resolved by training each layer of the denoising auto-encoder separately, and these layers are stacked based on the weights. This type of AE is called a Stacked Denoising Auto Encoder. The basic AE uses a sigmoid activation function in the hidden layer, which is not beneficial when working with non-stationary data inputs. The ESDAE model uses a wavelet activation function instead of a sigmoid, performing better on the cross-domain datasets.

Furthermore, over fitting problems in AE are handled using the cost function. The proposed ESDAE model consists of input, hidden and layer. The network traffic dataset is given to the input layer and passed through the hidden layer, which uses the Morlet transform as an activation function to localize the spatial and temporal properties of the data. The output of the hidden layer is fed to the output layer. The structure of ESDAE is represented in Figure 2.

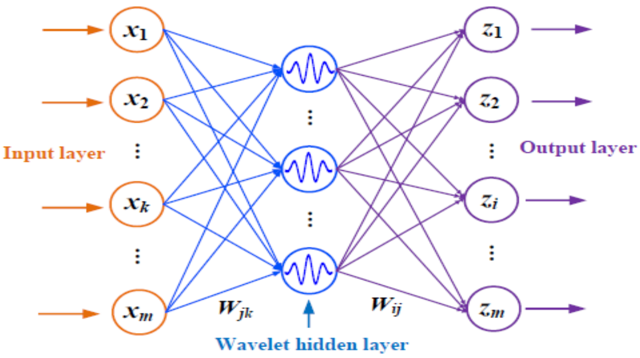

Figure 2. Structure of ESDAE

The input data of AE is taken as $x = [x_1, x_2, \ldots \ldots x_m]$ and it transforms the input using the activation function as the hidden feature vector $h = [h_1, h_2, \ldots h_p]$. The output points or the reconstruction vector is represented as $z = [z_1, z_2, \ldots \ldots z_m]$. The computation of hidden and output vectors are illustrated below,

$$\mathrm{h} = \mathrm{S_g}(\mathrm{Wx} + \mathrm{b}) \tag{7}$$

$$\mathrm{z} = \mathrm{S_f}(\mathrm{W'y} + \mathrm{b'}) \tag{8}$$

Where, S_g denotes the activation function of the hidden layer and S_f is the output layer's activation function. The sigmoid and Rectified Linear Unit (ReLU) is generally used as the activation function. $W$and W'denotes weights andb', b are the biases.

The cost function of the standard AE is specified as follows,

$$\mathrm{C_1} = \frac{1}{2}\sum_{\mathrm{i=1}}^{\mathrm{m}}(\mathrm{z_i} - \mathrm{x_i})^2 + \beta\left(\sum_{\mathrm{j=1}}^{\mathrm{p}} \mathrm{r}\log\frac{\mathrm{r}}{\hat{\mathrm{r}}_\mathrm{j}} + (1-\mathrm{r})\log\frac{1-\mathrm{r}}{1-\hat{\mathrm{r}}_\mathrm{j}}\right) \tag{9}$$

Here, β represents the sparse penalty coefficient r denotes the sparse constant. The proposed ESDAE method uses the Morlet wavelet activation function instead of sigmoid activation in the hidden layer because of its better performance in non-stationary data. The Morlet Wavelet is illustrated as,

$$\psi(\mathrm{t}) = \frac{1}{\sqrt{\mathrm{f_b}\pi}}\cos(2\pi\mathrm{f_c}\mathrm{t})\exp\left(\frac{-\mathrm{t}^2}{\mathrm{f_b}}\right) \tag{10}$$

Here f_b and f_c represents the bandwidth and central frequency parameters, which define the performance of the Morlet wavelet. With the implementation of Morlet wavelet as an activation function in the hidden layer, the output of the hidden layer is represented as follows,

$$\begin{gathered} h_j = \frac{1}{\sqrt{f_b\pi}} cos\big(2\pi f_c(\textstyle\sum_{k=1}^{m} W_{jk}x_k - c_j/d_j)\big). \\ exp\big(-((\textstyle\sum_{k=1}^{m} W_{jk}x_k - c_j)/d_j)^2/f_b\big) \end{gathered} \tag{11}$$

Here, hidden node j's output is specified as h_j, d_j represents the scalar factor, $c_j$represents the shift factor.W_{jk} represents the weight between the hidden node j and the input node k and W_{ij} represents the weight between the hidden node j and the output node i, respectively. Next, the nonlinear transformation of the output layer is set as tanh function, and it is expressed as

$$z_i = \tanh\left(\sum_{j=1}^{p} W_{ij} h_j\right) \tag{12}$$

Further, the weight decay term λ is added to Eq. (9) to limit the over fitting issue, and it is expressed as,

$$C^T = \frac{1}{2}\sum_{i=1}^{m}(z_i - x_i)^2 + \frac{\lambda}{2}\sum_{i,k=1}^{m}\sum_{j=1}^{p}\left((W_{ij})^2 + W_{jk})^2\right) + \beta\left(\sum_{j=1}^{p} r\log\frac{r}{\hat{r}_j} + (1-r)log\frac{1-r}{1-\hat{r}_j}\right) \tag{13}$$

The decay strategy introduces many connecting weights and reduces the sparsity. The quality of reconstruction and sparsity is enhanced by curtailing the negative weights with the assistance of non-negative constraints. The cost function with non-negative constraint is expressed as follows,

$$C^E = \frac{1}{2}\sum_{i=1}^{m}(z_i - x_i)^2 + \frac{\delta}{2}\sum_{L=1}^{2}\sum_{I=1}^{S_L}\sum_{J=1}^{S_{L+1}} G\left(W_{JI}^{(L)}\right) + \beta\left(\sum_{j=1}^{p} r\log\frac{r}{\hat{r}_j} + (1-r)log\frac{1-r}{1-\hat{r}_j}\right) \tag{14}$$

Where, $G\left(W_{JI}^{(L)}\right) = \begin{cases} \left(W_{JI}^{(L)}\right)^2 & W_{JI}^{(L)} < 0 \\ 0 & W_{JI}^{(L)} \geq 0 \end{cases}$

In Eq. (13), the second term represents the non-negative constraint, δ indicates the penalty coefficient, C^E denotes the enhanced cost function, and Lth layer node dimension is represented as S_L. To maintain the C^E value as a minimum, the training phases have to adjust the weight $W_{JI}^{(L)}$. The weights are updated utilizing gradient descent with back propagation. It is represented as below,

$$\mathrm{W}_{\mathrm{JI}}^{(\mathrm{L})} = \mathrm{W}_{\mathrm{JI}}^{(\mathrm{L})} - \eta\frac{\partial \mathrm{C}^{\mathrm{M}}}{\partial \mathrm{W}_{\mathrm{JI}}^{(\mathrm{L})}} \qquad \mathrm{L} = 1,2 \tag{15}$$

Where, $\frac{\partial \mathrm{C}^{\mathrm{M}}}{\partial \mathrm{W}_{\mathrm{JI}}^{(\mathrm{L})}} = \frac{\partial \mathrm{C}_1}{\partial \mathrm{W}_{\mathrm{JI}}^{\mathrm{L}}} + \delta \mathrm{g}\left(\mathrm{W}_{\mathrm{JI}}^{\mathrm{L}}\right)$ and $\mathrm{g}\left(\mathrm{W}_{\mathrm{JI}}^{\mathrm{L}}\right) = \begin{cases} \left(\mathrm{W}_{\mathrm{JI}}^{(\mathrm{L})}\right)^2 & \mathrm{W}_{\mathrm{JI}}^{(\mathrm{L})} < 0 \\ 0 & \mathrm{W}_{\mathrm{JI}}^{(\mathrm{L})} \geq 0 \end{cases}$

In the Eq. (15) ηdenotes the learning rate and$W_{JI}^{(1)} = W_{jk}$,$W_{JI}^{(2)} = W_{ij}$. From Eq. (10), the performance of the Morlet wavelet is highly dependent on f_b and f_c parameters. To optimize these parameters and create an adaptive Morlet wavelet, the Fruitfly Optimization Algorithm (FOA) is employed. The process begins with the initialization of the Morlet wavelet in the ESDAE model, which has already been trained using training instances. Validation instances are then provided as input, and the maximum number of epochs and the initial location of the fruitfly swarm are determined. Each fruitfly performs a random search for food based on its "smell" perception, involving a random search distance and direction. The smell concentration is calculated using the distance between each fruitfly and the origin. These values are then evaluated through a fitness function to identify the location with the minimum smell concentration. The fruitfly swarm saves this best smell concentration and adjusts its position, moving toward the optimal location using its vision. This process is repeated iteratively until the maximum number of epochs is reached.

Once the adaptive Morlet wavelet is formed, it is utilized within the ESDAE model to effectively eliminate noise from traffic data. This noise removal process enables the autoencoder to seamlessly extract both spatial and temporal features from the data, ensuring uninterrupted and efficient performance.

3.3. Cellular Traffic Prediction using MDSTS-CLSTM

After pre-processing, the network traffic data is given for feature extraction. The multi-dimensional learning approach is used for extracting the features, which is implemented in the CNN model, which extracts the spatial features. Then, these features are concatenated and passed through the LSTM model to extract the temporal features of network traffic data. Based on the extracted features, the classifier predicts the network traffic. The performance of CLSTM is enhanced by improving properties such as sparse representation, spatial, temporal analysis and multi-dimensional feature learning. The properties of the hybrid model and its involvement in traffic prediction are explained as follows.

Sparse Representation: The sparse transform is performed in one of the shallow layers of the CNN model, which is said to be the sparse representation layer. The sparse representation property enhances the ability of feature extraction and it is insensitive for the noisy data. It creates multiple numbers of feature maps, and it incorporates intrinsic characteristics of traffic data. The sparse representation is implemented utilizing wavelet and Shearlet transform methods The sparse representation layer could be placed in the shallow layer in many ways. It might be located in the front or back part of the network or the entire network. The sparse representation layer is included in front of each convolutional layer means the front part of the network provides better performance, and this placement method is taken in the proposed methodology. The network traffic data is given as input to the CNN model, including the sparse representation block. The convolution and sparse representation blocks process the input data. The sparse transform operation is performed to extract the features more efficiently in the sparse representation layer. No up dation is required for the sparse layer compared to the convolutional layer; it extracts both the spatial and temporal features using the sparse representation methods. The Extracted features are used by pooling and fully connected layers to produce the output. The sparse representation in CNN is displayed in Figure 3.

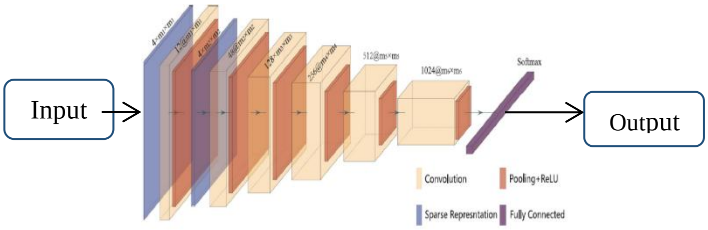

Figure 3. CNN with Sparse Representation Layer

This work uses a type of wavelet transform named Haar wavelet for sparse representation. The wavelet transforms are used for extracting the local features of the traffic data. The formulation of the Haar Wavelet transform is illustrated as,

$$\mathrm{Wf(a,b)} = \int_{-\infty}^{\infty} \mathrm{f(x)}\psi_{\mathrm{a,b}}\mathrm{(x)dx} = \langle \mathrm{f}, \psi_{\mathrm{a,b}} \rangle \tag{16}$$

Here, $f(x)$represents the traffic data, $\psi_{a,b}(x)$ indicates the wavelet function created from the mother wavelet function ψ.

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \tag{17}$$

Here, a and b represent the shift of scale and translation, respectively. However, Haar wavelets have some limitations in extracting multi-dimensional features. This limitation is resolved using the Shearlet transform. This method is useful for multi-scale geometric analysis and is derived using Fourier transform. The formulation of the Fourier transform is,

$\widehat{\Psi}(\xi) = \widehat{\Psi}(\xi_1, \xi_2) = \widehat{\Psi}(\xi_1)\widehat{\Psi}\left(\frac{\xi_1}{\xi_2}\right)for\forall_{\xi} = (\xi_1, \xi_2) \in R^2$,$\xi_1 \neq 0$. Then the Shearlet transform for the traffic data $f = L^2(R^2)$ derived as, (18)

$$SH_{\psi}f(a,s,t) = \langle f, \Psi_{a,s,t} \rangle$$

Here, $\psi_{a,s,t}(x) = \left|\det M_{a,s}\right|^{\frac{1}{2}}\psi\left(M_{a,s}^{-1}x - t\right)$;

$$M_{a,s} = \begin{vmatrix} a & \sqrt{as} \\ 0 & \sqrt{a} \end{vmatrix}$$

Then, the Shearlet transform is obtained as $\left\{\psi_{a,s,t}(x): a > 0, s \in R, t \in R^2\right\}$

Each matrix $M_{a,s}$is decomposed into the shear matrix, and the anisotropic dilation matrix is $B_s = \begin{vmatrix} 1 & s \\ 0 & 1 \end{vmatrix}$ and $A_a = \begin{vmatrix} a & 0 \\ 0 & \sqrt{a} \end{vmatrix}$. Hence, two kinds of operation, such as directional shearing and anisotropic dilation, are performed in $M_{a,s}$Matrix. The multi-scale analysis of shearlet transform is shown in Figure 4.

 ISSN 2723-6471

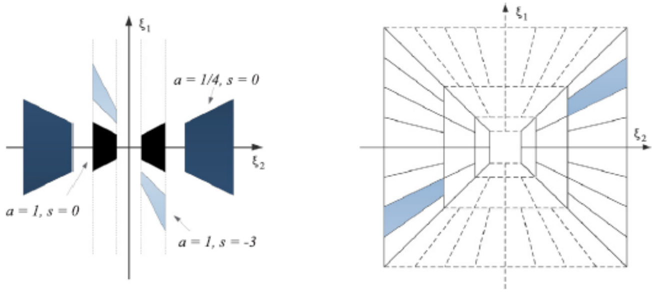

Figure 4. Shearlet transform

In Figure 4, the frequency domains of Shearlet at various scales are represented as trapeziums, which lie along the line with the slope k, and it is symmetrical about the origin. The shearlet transform has the local representation ability, which is determined by the parameters a, s and t. When the parameter a is decreased, it enables us to learn the multi-dimensional features. In the case of discrete shearlet transform, parameters$(a, s, t) \in R^{+} \times R \times R^{2}$ representing scaling, direction and translation are sampled.

The sparse representation property enhances the feature extraction process using wavelet and shearlet transforms. These two transforms help in extracting the multi-dimensional features of network traffic data. This property reduces the standard CNN model's computation cost and memory consumption.

3.3.1. Multi-dimensional Learning

The multi-dimensional learning approach is followed by the proposed model to extract the predominant features of the cellular traffic dataset. It uses multiple layers for extracting the essential features. The first layer independently learns the traffic data feature for each channel, and the second layer extracts the interactive features between the independent channels. Similarly, this operation is performed for n channels to extract the most beneficial features. While comparing, the feature map size of layer 1 is bigger than the other layers. The input traffic data incorporating SMS, call and internet traffic is initially given to the Conv-1layer. It is constructed with 50, 1×8 filters and is used in multi-dimensional learning procedures as input. In this phase, multiple convolutional layers are formed to predict input data traffic. The first layer, represented as Conv-2A to Conv-N+1A, extracts the features in the whole data, whereas the second layer that is Conv-3A to Conv-N+1A extracts the interactive features between the traffic data. Likewise, it performs up to n layers for extracting the essential spatial features. A feature concatenation procedure merges the learned features to categorize the information between the layers. The extracted features are given to the LSTM model for exploring the temporal feature representation. The extracted spatial and temporal feature of cellular traffic is sent to a fully connected layer for predicting the traffic. This consumes less time to learn the spatial features and increases the CTP system's speed. The multi-dimensional learning approach is illustrated in Figure 5.

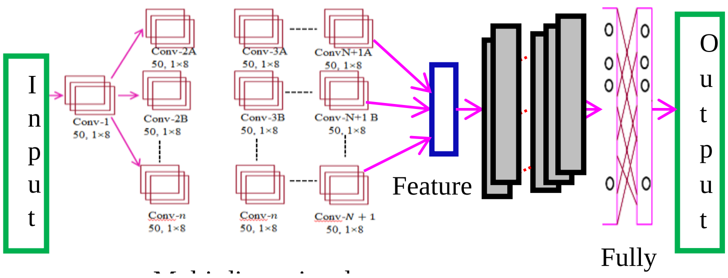

Figure 5. Representation of Multi-dimensional property

3.3.2. Spatial-Temporal Analysis

In the CLSTM model, the spatial features are learned using the CNN model, whereas the LSTM model learns the temporal features. The CNN model performs convolution and pooling operations to learn the spatial data and builds the complex high-dimensional matrix. The LSTM model can handle regular, periodic and time-series data. LSTM model uses the feature extracted by CNN and extracts the highly relevant features. This hybrid model produces more

ISSN 2723-6471

reliable and stable results in CTP. This model incorporates convolution, dropout and max pooling layer. The multi-dimensional traffic data is fed as an input to the convolutional layer, which utilizes a sliding window to compress the data. Then, the irrelevant features are filtered using the dropout layer. The filtered features are given to the max pooling layer, which effectively extracts the spatial features. Lastly, the LSTM model extracts the temporal features of traffic data. The spatial and temporal feature analysis process is displayed in Figure 6.

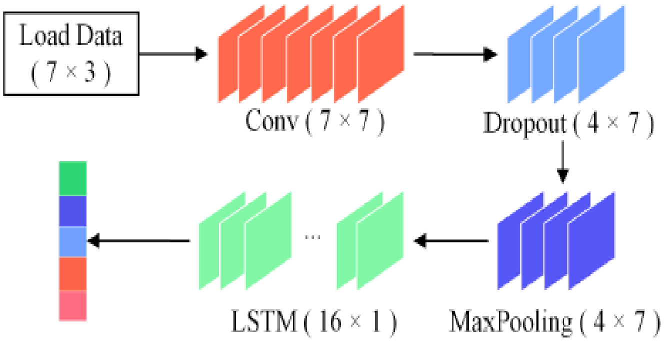

Figure 6. Process in spatial and temporal feature analysis

This hybrid model predicts the network traffic data's spatial and temporal relationship in a limited period. It produces better outcomes using a limited number of parameters.

3.3.3. MDSTS-CLSTM

The cross-domain cellular traffic data contains noise, affecting the performance of the traffic prediction model. Hence, it is filtered using ESDAE. Further, the denoised data is input to the MDSTS-CLSTM model which predicts the cellular traffic. The denoised traffic data is passed to the convolution layer, which learns the features of traffic data. The learned features are given to the sparse representation layer, which extracts the spatial features using wavelet and shearlet transforms. Both the transforms are used to extract the spatial and temporal features. Further, the multi-dimensional property extracts the spatial representations using the multiple layers. These features are concatenated and given to the LSTM block, extracting the temporal representation of the traffic data. Then, the spatial and temporal features are fed to the fully connected layer, which predicts the cellular network traffic and generates the outcome. The architecture of MDSTS-CLSTM is represented in Figure 7.

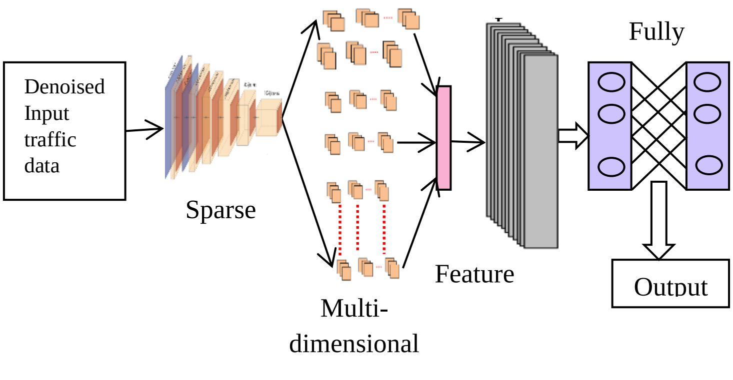

Figure 7. Architecture of MDSTS-CLSTM

The CNN models are used to extract the spatial feature, which is ineffective in extracting the temporal features. Hence, the LSTM model is combined with CNN to extract the temporal features. This combination effectively learns the network traffic data's spatial and temporal features. The problem of spatial invariance is solved using the convolution operation instead of the dot product in the standard LSTM model. The LSTM models have three gates such as input gate i, forget gate f and output gate o. The forget gate is used to remove or store the information belonging to the memory cell c. The basic LSTM models are one-dimensional, using a single cell and incorporating a single recurrent connection. The activation of the recurrent connection is controlled utilizing the single forget gate. In the case of multi-dimensional LSTM, it uses multiple cells and the recurrent connection. To model the LSTM, the inputs are taken as $x_1, x_2, x_3 \ldots\ldots x_t$, output as $c_1, c_2, c_3 \ldots\ldots c_t$ and the hidden state as $h_1, h_2, h_3 \ldots\ldots h_t$. The main operation of the suggested CLSTM model is expressed as below,

$$\mathrm{i_t} = \sigma(\mathrm{W_{xi}} * \mathrm{x_t} + \mathrm{W_{hi}} * \mathrm{h_{t-1}} + \mathrm{W_{ci}} \circ \mathrm{c_{t-1}} + \mathrm{b_i} \tag{19}$$

$$\mathrm{f_t} = \sigma(\mathrm{W_{xf}} * \mathrm{x_t} + \mathrm{W_{hf}} * \mathrm{h_{t-1}} + \mathrm{W_{cf}} \circ \mathrm{c_{t-1}} + \mathrm{b_f} \tag{20}$$

$$\mathrm{o_t} = \sigma(\mathrm{W_{xo}} * \mathrm{x_t} + \mathrm{W_{ho}} * \mathrm{h_{t-1}} + \mathrm{W_{co}} \circ \mathrm{c_{t-1}} + \mathrm{b_o}) \tag{21}$$

$$\mathrm{c_t} = \mathrm{f_t} \circ \mathrm{c_{t-1}} + \mathrm{i_t} \circ \tanh(\mathrm{W_{xc}} * \mathrm{x_t} + \mathrm{W_{hc}} * \mathrm{h_{t-1}} + \mathrm{b_c}) \tag{22}$$

$$\mathrm{h_t} = \mathrm{o_t} \circ \tanh(\mathrm{c_t}) \tag{23}$$

Here, $\circ$ denotes the Hadamard product and $*$ represents the convolutional operator. The LSTM model includes five layers, three hidden layers, and two feed-forward subsampling layers. The input data is given to the hidden layer using the input layer, and the data is processed and sent to the output layer. The subsampling layers use $\tanh$ activation function, and this layer fastens the training time by compressing the sequences as windows. This layer reduces the weight connection between hidden layers. Furthermore, the LSTM model is trained using Connection Temporal Classification in the output layer, increasing the probability of labeling sequence during training.

In this work, some LSTM parameters are tuned during training. The parameters are the size of LSTM, $tanh$ and the subsampling layer. The LSTM size is the number of cells in each hidden layer, and the size is taken as 2, 10 and 5. The $\tanh$ size represents the used $tanh$ units in the subsampling layer. The subsampling window size indicates the window used to subsample the input before giving it to the coming hidden layers.

4. Results and Discussion

Extensive experiments are performed using MATLAB to validate the effective performance of the proposed MDSTS-CLSTM-based traffic prediction model. The performance of the model is evaluated in terms of accuracy, precision, recall, false positive rate (FPR), false negative rate (FNR), mean absolute error (MAE), and RMSE. The Milan city dataset is used along with the cross-domain datasets for evaluation. Table 1 shows the average results obtained for the proposed model over different service datasets of the network traffic.

Table 1. Performance of proposed ICTPM using ESDAE and MDSTS-CLSTM

Dataset	Type	Accuracy	RMSE	MAE
SMS	No cross-domain	0.8388	45.76	31.54
	+Social	0.8656	47.65	30.54
	+BS	0.8675	48.55	26.22
	+POI	0.8703	49.12	28.65
Call	No cross-domain	0.8577	41.45	15.43
	+Social	0.8772	38.65	16.54
	+BS	0.8898	39.41	15.76
	+POI	0.9054	38.86	13.51
Internet	No cross-domain	0.9234	65.67	75.78
	+Social	0.9143	66.20	72.34
	+BS	0.9367	68.56	73.21
	+POI	0.9667	60.32	70.43

From Table 1, it is shown that the proposed ESDAE and MDSTS-CLSTM methods have significantly improved the network traffic predictions. The noise removal and the enhanced handling of the sparse data can be attributed to this improvement. In addition, the proposed ICTPM is also compared against the existing methods from the literature to identify its effectiveness. Table 2 shows the average results obtained for the proposed model against the existing methods under similar experimental conditions.

Table 2. Performance Comparison of ICTPM against Existing Methods

Methods	Accuracy	Precision	Recall	FPR	FNR	MAE	RMSE
STCNet [5]	0.8150	0.9123	0.8454	0.2422	0.3345	29.87	60.32
MGCN-LSTM [6]	0.8619	0.9067	0.8765	0.2805	0.3012	36.67	66.41
STGCGRU [7]	0.7754	0.9112	0.7912	0.1654	0.2987	33.21	58.23
CNN [8]	0.8004	0.9687	0.8319	0.1320	0.1876	38.76	59.11
MVSTGN [9]	0.8602	0.9701	0.8495	0.1988	0.2134	29.98	49.03
A-MCSTCNet [10]	0.8933	0.9323	0.7663	0.2023	0.2976	23.30	51.93
EDRL [11]	0.8492	0.9651	08231	0.2250	0.2341	21.34	60.02
CrowdGAN [12]	0.8110	0.9011	0.8700	0.1991	0.2765	25.55	54.45
SDGNet [13]	0.8667	0.8976	0.8799	0.1976	0.1987	31.76	49.91
SLSTM [14]	0.8725	0.8851	0.9023	0.2419	0.2123	33.34	53.36
1DCNN-GRU [15]	0.8898	0.9550	0.9102	0.1325	0.1765	32.92	63.30
GRU [16]	0.8518	0.8932	0.8543	0.1559	0.2098	29.68	47.65
CNN-LSTM [17]	0.8334	0.9222	0.8698	0.1287	0.1876	23.74	51.49
GCN [18]	0.7932	0.9007	0.8324	0.1832	0.2567	29.81	50.35
GCNN-LSTM [19]	0.8876	0.9665	0.8712	0.2102	0.2987	24.49	54.48
RL [20]	0.8654	0.9571	0.9011	0.1450	0.1902	23.41	49.33
Proposed ICTPM	0.9036	0.9797	0.9175	0.1128	0.1616	20.67	45.46

From Table 2, it is evident that the proposed ICTPM using ESDAE and MDSTS-CLSTM methods have better performance in terms of higher accuracy, precision, recall, and reduced FPR, FNR, MAE, and RMSE, which are better than the existing methods discussed in the literature. The significant improvement of the proposed ICTPM shows that it is effective for adaptive learning of the features in accurately predicting the network traffic. It is also indicative that the ICTPM can converge better than the other model with minimized noise and minimized model complexity.

5. Conclusion

In this paper, an intelligent cellular traffic prediction model (ICTPM) leverages two improved deep learning algorithms,ESDAE and MDSTS-CLSTM,to handle the challenges of noisy and sparse traffic data from cross-domains. The ESDAE employs an adaptive Morlet wavelet transform to remove the noise from the traffic data and enhance the data quality. The MDSTS-CLSTM combines LSTM and CNN's advantages to learn the traffic data's hidden patterns and dependencies in both spatial and temporal dimensions. The MDSTS-CLSTM also incorporates a sparse representation technique to deal with the data sparsity problem and reduce computation time and storage costs. The experimental results show that the proposed ICTPM achieves higher accuracy, precision, recall, and reduced FPR, FNR, MAE, and RMSE ensuring robustness than the existing methods. The proposed ICTPM model has demonstrated its effectiveness and superiority for cellular traffic prediction using cross-domain datasets. However, some limitations and challenges still need to be addressed in future research. Extending ICTPM to handle multi-domain and multi-source traffic data, such as social media, video streaming, and IoT devices, and exploring the potential correlations and interactions among different types of traffic is a future research suggestion.

6. Declarations

6.1. Author Contributions

Conceptualization: A.S.S., N.S., S.S.M., Q.Y.; Methodology: S.S.M.; Software: A.S.S.; Validation: A.S.S., S.S.M., dan Q.Y.; Formal Analysis: A.S.S., S.S.M., dan Q.Y.; Investigation: A.S.S.; Resources: S.S.M.; Data Curation: S.S.M.;

Writing Original Draft Preparation: A.S.S., S.S.M., dan Q.Y.; Writing Review and Editing: S.S.M., A.S.S., dan Q.Y.; Visualization: A.S.S.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (NTMA): A survey," *Computer Communications*, vol. 170, no. 1, pp. 19–41, 2021.

[2] W. Jiang, "Cellular traffic prediction with machine learning: A survey," *Expert Systems with Applications*, vol. 2022, no. 1, pp. 1–15, 2022.

[3] Lohrasbinasab, A. Shahraki, A. Taherkordi, and A. Delia Jurcut, "From statistical to machine learning-based network traffic prediction," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, pp. 1–10, 2022.

[4] K. Lee, M. Eo, E. Jung, Y. Yoon, and W. Rhee, "Short-term traffic prediction with deep neural networks: A survey," *IEEE Access*, vol. 9, no. 5, pp. 54739–54756, 2021.

[5] 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, 2019.

[6] J. Lin, Y. Chen, H. Zheng, M. Ding, P. Cheng, and L. Hanzo, "A data-driven base station sleeping strategy based on traffic prediction," *IEEE Transactions on Network Science and Engineering*, vol. 2021, no. 8, pp. 1–10, 2021.

[7] K. Zhang, X. Zhao, X. Li, X. You, and Y. Zhu, "Network traffic prediction via deep graph-sequence spatiotemporal modeling based on mobile virtual reality technology," *Wireless Communications and Mobile Computing*, vol. 2021, no. 9, pp. 1–7, 2021.

[8] W. Shen, H. Zhang, S. Guo, and C. Zhang, "Time-wise attention-aided convolutional neural network for data-driven cellular traffic prediction," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1747–1751, 2021.

[9] Y. Yao, B. Gu, Z. Su, and M. Guizani, "Mvstgn: A multi-view spatial-temporal graph network for cellular traffic prediction," *IEEE Transactions on Mobile Computing*, vol. 2021, no. 10, pp. 1–10, 2021.

[10] Q. Zeng, Q. Sun, G. Chen, and H. Duan, "Attention-based multi-component spatiotemporal cross-domain neural network model for wireless cellular network traffic prediction," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 5, pp. 1–25, 2021.

[11] N. M. Balamurugan, M. Adimoolam, M. H. Alsharif, and P. Uthansakul, "A novel method for improved network traffic prediction using enhanced deep reinforcement learning algorithm," *Sensors*, vol. 22, no. 13, pp. 1–12, 2022.

[12] J. H. Duan, W. Li, X. Zhang, and S. Lu, "Forecasting fine-grained city-scale cellular traffic with sparse crowdsourced measurements," *Computer Networks*, vol. 214, no. 2, pp. 1–8, 2022.

[13] Y. Fang, S. Ergüt, and P. Patras, "SDGNet: A Handover-Aware Spatiotemporal Graph Neural Network for Mobile Traffic Forecasting," *IEEE Communications Letters*, vol. 26, no. 3, pp. 582–586, 2022.

[14] Z. Gao, "5G traffic prediction based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, no. 6, pp. 1–12, 2022.

[15] C. H. Lin and U. Nuha, "RobustSTL and machine-learning hybrid to improve time series prediction of base station traffic," *Electronics*, vol. 11, no. 8, pp. 1–10, 2022.

[16] Kumar, R. S. Umurzoqovich, N. D. Duong, P. Kanani, A. Kuppusamy, and M. Praneesh, "An intrusion identification and prevention for cloud computing: From the perspective of deep learning," *Optik*, vol. 270, no. 11, pp. 1–12, 2022.

[17] K. Selvi and R. Thamilselvan, "An intelligent traffic prediction framework for 5G network using SDN and fusion learning," *Peer-to-Peer Networking and Applications*, vol. 15, no. 1, pp. 751–767, 2022.

[18] S. Shawel, E. Mare, T. T. Debella, S. Pollin, and D. H. Woldegebreal, "A multivariate approach for spatiotemporal mobile data traffic prediction," *Engineering Proceedings*, vol. 18, no. 1, pp. 1–5, 2022.

[19] X. Zhou, Y. Zhang, Z. Li, X. Wang, J. Zhao, and Z. Zhang, "Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning," *Neural Computing and Applications*, vol. 34, no. 7, pp. 5549–5559, 2022.

[20] G. Chen, Y. Guo, Q. Zeng, and Y. Zhang, "A novel cellular network traffic prediction algorithm based on graph convolution neural networks and long short-term memory through extraction of spatial-temporal characteristics," *Processes*, vol. 11, no. 8, pp. 1–10, 2023.

[21] L. Nie, X. Wang, Q. Zhao, Z. Shang, L. Feng, and G. Li, "Digital twin for transportation big data: A reinforcement learning-based network traffic prediction approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2023, no. 1, pp. 1–10, 2023.