*Intelligent and Converged Networks* ISSN 2708-6240
2025, 6(1): 82−101 DOI: 10.23919/ICN.2025.0005

# DSTL: A dual-step transfer learning-based prediction model for next-generation intelligent cellular networks

**Waqar A. Aziz*, Iacovos I. Ioannou, Marios Lestas, and Vasos Vassiliou**

**Abstract:** Traffic modeling and prediction are indispensable to future extensive data-driven automated intelligent cellular networks. It contributes to proactive and autonomic network control operations within cellular networks. Current methodologies typically rely on established prediction models designed for univariate and multivariate time series forecasting. However, these approaches often demand a substantial volume of training data and extensive computational resources for prediction model training. In this study, we introduce a dual-step transfer learning (DSTL)-based prediction model specifically designed for the prediction of multivariate spatio-temporal cellular traffic. This technique involves the categorization of gNodeBs (gNBs) into distinct clusters based on their traffic pattern correlations. Instead of training the prediction model individually on each gNB, a base model is trained on the aggregated dataset of all the gNBs within a base cluster using a combination of recurrent neural network (RNN) and bidirectional long-short term memory (RNN-BLSTM) network. In the first-step transfer learning (TL), the base model is provided to the gNBs within the base cluster and to the other clusters, where it undergoes the process of fine-tuning the intra-cluster aggregated dataset. Once the model is trained on the aggregated dataset within each cluster, it is provided to the gNBs within the respective cluster in the second-step TL. The model received by each gNB through the proposed DSTL technique either necessitates minimal fine-tuning or, in some cases, requires no further adjustment. We conduct extensive experiments on a real-world Telecom Italia cellular traffic dataset. The results demonstrate that the proposed DSTL-based prediction model achieves a mean absolute percentage error of 2.97%, 9.85%, and 9.73% in predicting spatio-temporal Internet, calling, and messaging traffic, respectively, while utilizing less computational resources and requiring less training time than traditional model training and TL techniques.

**Key words:** dual-step transfer learning (DSTL); multivariate spatio-temporal cellular traffic prediction; bidirectional long-short term memory (RNN-BLSTM)

## 1 Introduction

Modeling and predicting traffic patterns are integral components of future automated intelligent cellular networks[1] driven by big data[2] since they play a critical role in enabling autonomic network control[3], effective management[4], and seamless service provisioning[5]. Specifically, the efficiency of the proactive demand-aware resource allocation largely benefits from the accurate prediction of future wireless traffic[6, 7]. Cellular traffic primarily consists of short message service (SMS), call, and Internet traffic[8]. However, predicting cellular traffic remains challenging due to its diverse and complex multivariate spatio-temporal traffic patterns[9].

The prediction methods can be broadly categorized into two groups: statistical-based and machine learning (ML)-based methods[10]. The first category involves

• Waqar A. Aziz, Iacovos I. Ioannou, and Vasos Vassiliou are with Department of Computer Science, University of Cyprus, Nicosia 2109, Cyprus, and also with CYENS Centre of Excellence, Nicosia 1076, Cyprus. E-mail: w.a.aziz9@outlook.com; ioannou.iakovos@ucy.ac.cy; vassiliou.vasos@ucy.ac.cy.
• Marios Lestas is with Department of Electrical Engineering, Frederick University, Nicosia 1036, Cyprus. E-mail: eng.lm@frederick.ac.cy.
∗ To whom correspondence should be addressed.
An earlier version of this paper was presented in part at the IEEE ICC 2023 DDINS Workshop.
Manuscript received: 2023-11-19; revised: 2024-08-29; accepted: 2024-11-12

modeling and predicting cellular traffic using statistics or probabilistic distributions[11], which includes $\alpha$-stable distribution[12], auto-regressive integrated moving average (ARIMA)[13–15], covariance function[16], and entropy theory[17]. The second category encompasses ML[18] and artificial intelligence (AI) techniques[19]. Data-driven ML-based regression models have emerged as robust contenders to classical statistical prediction models in the field of wireless communication[20]. Initially, several shallow learning methods, including linear regression[21] and support vector regression (SVR)[22], were employed for traffic prediction. However, the advancement in deep learning (DL) has led to the adoption of more sophisticated regression models such as recurrent neural network (RNN)[23] and long short-term memory (LSTM) networks[24]. These DL models have filled the gap and demonstrated substantial improvements in cellular traffic prediction.

The dynamic nature of network conditions, such as varying traffic loads, interference, and user mobility, complicates the ability of traditional ML-based regression models to generalize across diverse environments[25]. Furthermore, the performance of ML-based regression models mainly depends on the availability of adequate training data. Acquiring a sufficient amount of data might be costly and time-consuming[26]. Besides, if the training data are sufficient, regression models, such as DL, usually require a long training time, which makes them impractical for many latency-sensitive applications, i.e., cellular traffic prediction.

To address these challenges, transfer learning (TL) has recently emerged as a promising solution by transferring the knowledge of a pre-trained model from source to a target domain[9, 27]. The effectiveness of the TL method is contingent upon the feature representations extracted from the data being pertinent to both the source and target tasks[28]. TL yields optimal results when the traffic patterns in the source and target domains closely align. In the context of cellular traffic for gNodeBs (gNBs), the feature set exhibits similarities across gNBs, with the exception of the diverse traffic patterns. Applying a pre-trained model's knowledge to a negatively correlated traffic pattern is akin to training an ML model from scratch, which diminishes the benefits of TL. Therefore, a thorough analysis of the target domain's traffic pattern is essential before implementing TL[29].

This paper proposes an optimal TL technique to address the challenges of ML model training in cellular networks. The classical TL-based prediction model is employed for target gNBs exhibiting comparable traffic patterns. In contrast, a dual-step transfer learning (DSTL)-based prediction model is applied to the target gNBs that demonstrate a lower correlation in traffic patterns than the source gNB[30]. To streamline the process, gNBs are categorized into distinct clusters based on historical traffic pattern correlations. Addressing the challenge of training the ML model on big data at each gNB individually, the initial base model is trained on the edge server within a cluster. Instead of combining data samples from all the gNBs, the data samples are aggregated on a per-time basis across the participating gNBs, creating a unified traffic pattern for the initial base model, comprising the traffic patterns of all the gNBs within the cluster. This cluster, termed a base cluster, contains the initial base model. The base model is then transferred to the gNBs within the base cluster using an inductive transfer learning (ITL) approach. However, due to the lower correlation with the gNBs residing in the other clusters, this base model yields lower prediction accuracy, necessitating the exploration of excessive hyperparameter combinations to fine-tune the base model on the local traffic patterns of each gNB. Alternately, the base model can be first fine-tuned on the aggregated dataset of other clusters. In the second step TL, the fine-tuned model is transferred to the gNBs of the respective clusters. This DSTL technique mitigates the need for extensive exploration of hyperparameter combinations for ML model training in each gNB, thus minimizing computational resources and reducing ML model training time across all gNBs.

Following the proposed framework and description provided above, our contributions to this paper can be summarized as follows:

(1) Leveraging a sophisticated clustering strategy, we

categorize gNBs into different clusters based on the correlations of their traffic patterns.

(2) By aggregating traffic patterns across participating gNBs on a per-time basis, we streamline the training process of the initial base model. This contributes to a multi-fold reduction in base model training time, particularly in the context of large-scale cellular network data.

(3) To address the challenge of lower correlation in traffic patterns between clusters, we propose a novel DSTL technique. This technique mitigates the necessity for an exhaustive exploration of hyperparameter combinations during ML model training in each gNB, thereby reducing both computational resources and ML model training time.

(4) We propose utilizing an advanced RNN and bidirectional LSTM (RNN-BLSTM) neural network architecture tailored explicitly for predicting multivariate spatio-temporal cellular traffic. This architecture enhances the model's predictive capabilities on the sequential datasets, contributing to more accurate and reliable predictions.

The paper utilizes a dataset generated by the Telecom Italia cellular network from the city of Milan[31]. The cellular coverage in Milan is structured into cells, forming a $100 \times 100$ grid. Each cell corresponds to a specific region where user telecommunication activities are served and logged by the gNB. In this context, the terms "gNB" and "cell" are used interchangeably[32].

The subsequent sections of the paper are organized as follows: Section 2 provides an overview of the related work. Section 3 presents a concise formulation of the problem, and Section 4 delves into a detailed description of the proposed framework. Section 6 offers insights into the results based on the experimental setup outlined in Section 5. The paper's conclusion is presented in Section 7.

## 2 Related work

### 2.1 Cellular traffic prediction

The literature shows that a diverse array of prediction models can be applied to forecast time series distributions. Statistical models such as ARIMA[13–15] and Holt-Winter (HW)[33] are notable examples. These models demand less computation, making them ideal for environments with limited storage and computational resources. However, ML models tend to outperform statistical models in scenarios with a nonlinear relationship between input and output values. ML models encompass a variety of techniques, including random forest[34], Gaussian process regression[35], linear regression[21], SVR[22], and Prophet[36].

In the context of advanced prediction techniques, DL models stand out. These encompass sophisticated approaches like feed-forward neural network (FFNN)[37], convolutional neural network (CNN)[38], graph neural network (GNN)[39], and RNN[23]. Variants of RNNs, such as LSTM[24] and gated recurrent unit (GRU)[40], were widely employed in the literature. For a more comprehensive overview of cellular traffic prediction models, readers are referred to Ref. [9].

### 2.2 Transfer learning

In TL, a pre-trained source model is utilized in a target domain. TL is often integrated with machine learning algorithms, including deep neural networks. The primary objective of TL is to generate an efficient model for a target domain with a scarcity of labeled data samples while concurrently minimizing computation costs[41]. Specifically, the parameter-based TL approach is frequently employed in time series prediction problems, where knowledge transfer occurs at the model/parameter level[42]. TL can be crucial in facilitating new gNB deployments by leveraging historical data from existing gNBs through deep TL[43].

In Ref. [44], the authors proposed the utilization of CNN in conjunction with TL to enhance prediction accuracy. However, it was well established in the literature that time series prediction problems are better addressed by RNNs rather than CNNs. In Ref. [45], CNN was employed to model spatial dependencies, while the LSTM network handled the time series aspects of cellular traffic. The authors introduced an inter-clustering TL strategy to analyze and leverage similarities in both spatial and temporal domains.

Similarly, Ref. [46] employed CNN for cell classification and LSTM for training the prediction model. In Ref. [47], a convolutional LSTM (convLSTM) neural network was used for model training, and the Pearson correlation coefficient was employed to assess the correlation between cross-domain datasets and diverse cellular traffic. Reference [48] adopted an inter-cluster TL approach for model training, incorporating a generative adversarial network (GAN) for domain adaptation between the source and target data. In Ref. [49], a regularization-based TL method was employed for knowledge sharing with the target domain. However, this approach necessitated using cloud servers to store data from all gNBs in the area of interest, imposing an increased burden on the network side and demanding substantial cloud resources for data storage and model training. Reference [50] applied an LSTM-based TL approach designed explicitly for univariate data. In Ref. [51], the objective was to transfer the knowledge gained from a general ML model trained with general data and subsequently retrain the ML model on a more specific local gNB dataset. Reference [52] utilized the domain similarity between source and target cells to select the most appropriate cell as the source domain for the TL task.

The methodologies outlined in the approaches above are constrained to a singular input feature, specifically Internet traffic. This limitation arises from the inherent heterogeneity of multivariate spatio-temporal traffic, necessitating the implementation of a more intricate prediction model. Additionally, authors in Ref. [52] performed domain similarity analysis, but it lacked a solution for situations where the target domain exhibits different traffic patterns than the source domain. Similarly, authors in Ref. [51] introduced the concept of training a prediction model on a generic dataset to reduce computational resources on each gNB. Still, it lacked information about the correlation of the generic dataset with the dataset of each target gNB.

In our paper, we propose an innovative approach that involves inductive DSTL coupled with the fusion of an RNN-BLSTM network. Our technique initiates by categorizing gNBs into different clusters based on the similarity in traffic patterns. Subsequently, the prediction model undergoes training on the aggregated dataset within a cluster. The trained model is then shared with the target cluster and fine-tuned on its aggregated dataset. Once the model is generated, it can be deployed to gNBs for cellular traffic prediction with minimal processing overhead on the gNBs. This DSTL technique ensures enhanced accuracy in predicting multivariate spatio-temporal traffic within cellular networks while optimizing computational resources.

## 3 Problem formulation

Let's consider a scenario illustrated in Fig. 1. In this scenario, we have a source cluster with an aggregated dataset $\overline{X}_t$, and $\overline{X}_t^{(k)}$ represents the aggregated datasets of the target clusters. The datasets of the target gNBs within each cluster are denoted as $X_t^{(k)n}$. The primary aim is to develop a base model for a source cluster $\{M_{\mathrm{B}}\}$ and $k$ transferred models for the target clusters $\{M_{\mathrm{B}}^k\}$. Upon receiving the transferred model as a result of the first-step TL $\{M_{\mathrm{B}}^k\}$, the target cluster $k$ fits the model to its aggregated dataset, generating $n$ transferred models for the second-step TL for $n$ gNBs $\{M_{\mathrm{B}}^{(k)n}\}$ within the cluster. When new data sample $X_{(t+1)}^{(k)n}$ arrives at the $n$-th gNB within cluster $k$, the model must predict $Y_{(t+1)}^{(k)n}$ as accurately as possible. Therefore, the loss function can be expressed as

$$L^{(k)n} = \frac{1}{\left|M_{\mathrm{B}}^{(k)n}\right|} \sum_{Y_{(t+1)}^{(k)n} \in X_t^{(k)n}} d\left(\hat{Y}_{(t+1)}^{(k)n}, Y_{(t+1)}^{(k)n}\right) \tag{1}$$

where $d$ represents the error metric, precisely the mean absolute percentage error (MAPE), $\left|M_{\mathrm{B}}^{(k)n}\right|$ is the size of the dataset, and $\hat{Y}_{(t+1)}^{(k)n}$ is the prediction of the actual value $Y_{(t+1)}^{(k)n}$. The objective is to minimize this loss function while also considering minimising computational resources.

## 4 Proposed framework

This section gives a detailed description of our proposed multivariate spatio-temporal cellular traffic prediction framework. In the system model, for simulation purposes, we consider a number of clusters, $k = 3$. The system diagram with nine gNBs divided into three clusters is shown in Fig. 1. Our proposed

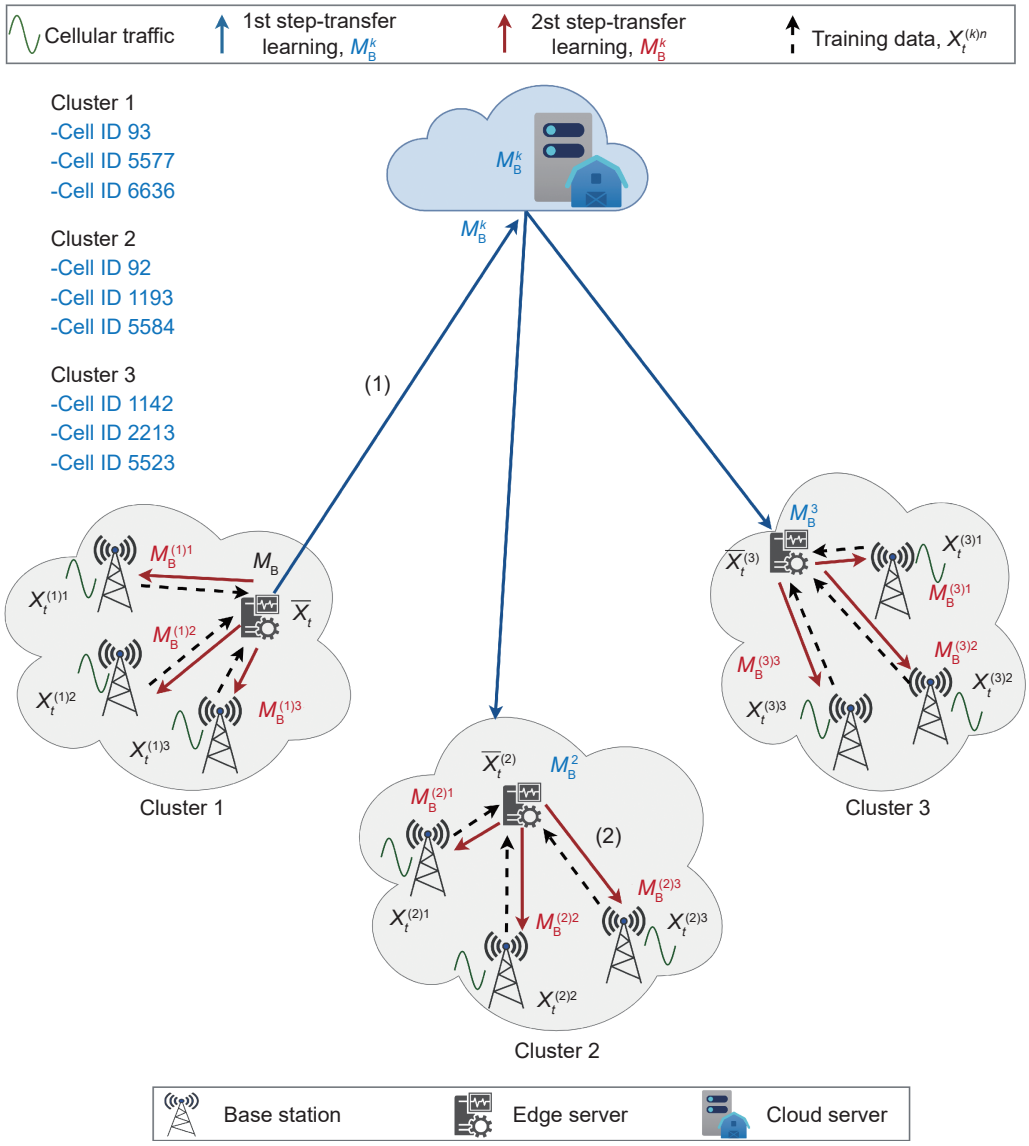



**Fig. 1 Illustration of the DSTL-based prediction model in a cellular network with clusters based on the similarity in traffic distribution among gNBs.**

framework is mainly comprised of four important steps:

(1) Segment a cellular network into different clusters based on the traffic pattern diversity and similarity.

(2) Data augmentation is performed within clusters.

(3) RNN-BLSTM neural network architecture.

(4) A cluster yields an intra-cluster RNN-BLSTM prediction model, which is transferred to other clusters using the dual-step TL technique.

### 4.1 Cellular network clustering

The distribution of cellular traffic across various cells reveals that traffic patterns are subject to factors, including the diverse requirements of mobile users at different times and locations and the spatial dependencies among gNBs created by user mobility. Consequently, developing an accurate prediction model necessitates a substantial dataset for training, particularly in the context of multivariate spatio-temporal traffic patterns.

In order to comprehend the varying traffic patterns across different gNBs, we can gain insight from the heatmap representation of geographically located gNBs, as illustrated in Fig. 2. The similarity in traffic patterns is quantified using the Pearson correlation coefficient. Based on the degree of similarity in traffic

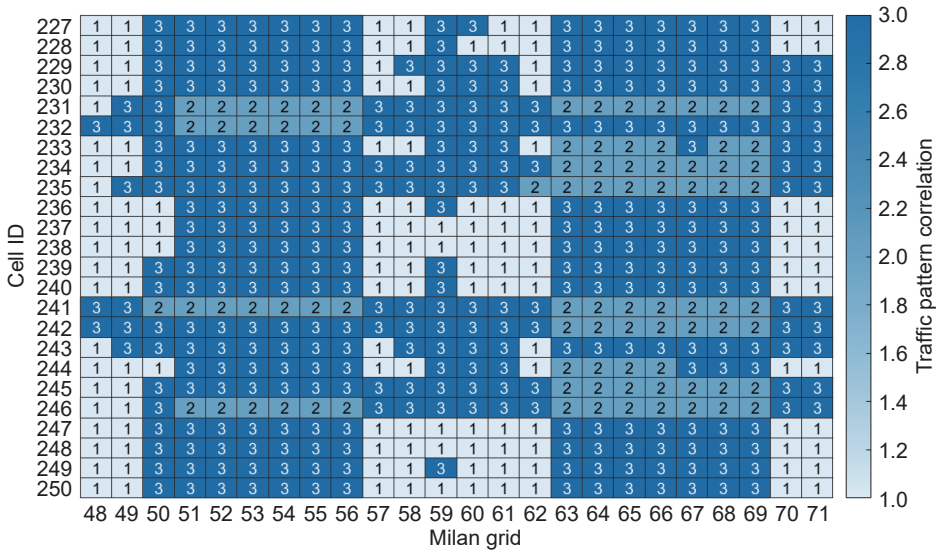



**Fig. 2 Categorization of the Telecom Italia cellular grid into appropriate clusters based on the correlation of traffic patterns.**

patterns, the gNBs are classified into three distinct clusters, each depicted by a different color in the heatmap. Building upon the rationale mentioned above, for the purposes of simulation, nine gNBs are selected due to their unique and diverse cellular traffic patterns, as visualized in Fig. 1.

The mutual correlation among the selected gNBs is depicted in Fig. 3. The correlation coefficient falls within the range of −1.000 to 1.000. A correlation of 1.000 signifies a perfect positive correlation, while −1.000 indicates a perfect negative correlation. The gNBs within a cluster exhibit stronger correlations with each other than those gNBs in other clusters. Let's consider the specific case of gNB 6636 in Cluster 1. However, it has correlations of 0.600, 0.580, and 0.570 with gNBs 92, 1193, and 5584 in Cluster 2, respectively, which is higher than the average correlation of gNBs in Cluster 2. However, it shows even higher correlations with gNBs 93 and 5577, which are 0.710 and 0.700, respectively. Consequently,

| gNB | 93 | 5577 | 6636 | 92 | 1193 | 5584 | 1142 | 2213 | 5523 |
|---|---|---|---|---|---|---|---|---|---|
| 93 | 1.000 | 0.700 | 0.710 | 0.570 | 0.440 | 0.290 | −0.110 | −0.100 | −0.027 |
| 5577 | 0.700 | 1.000 | 0.770 | 0.490 | 0.410 | 0.420 | 0.360 | 0.300 | 0.280 |
| 6636 | 0.710 | 0.770 | 1.000 | 0.600 | 0.580 | 0.570 | 0.350 | 0.310 | 0.390 |
| 92 | 0.570 | 0.490 | 0.600 | 1.000 | 0.580 | 0.540 | −0.110 | −0.120 | −0.035 |
| 1193 | 0.440 | 0.410 | 0.580 | 0.580 | 1.000 | 0.580 | 0.340 | 0.013 | 0.036 |
| 5584 | 0.290 | 0.420 | 0.570 | 0.540 | 0.580 | 1.000 | 0.017 | 0.080 | 0.062 |
| 1142 | −0.110 | 0.360 | 0.350 | −0.110 | 0.340 | 0.017 | 1.000 | 0.730 | 0.760 |
| 2213 | −0.100 | 0.300 | 0.310 | −0.120 | 0.013 | 0.080 | 0.730 | 1.000 | 0.760 |
| 5523 | −0.027 | 0.280 | 0.390 | −0.035 | 0.036 | 0.062 | 0.760 | 0.760 | 1.000 |

Cluster 1 Cluster 2 Cluster 3

**Fig. 3 Illustration of the intra-cluster and inter-cluster traffic pattern similarities among the gNBs, with each cluster represented by a different color.**

it is placed in Cluster 1 alongside gNBs 93 and 5577.

### 4.2 Prediction model

RNNs are commonly used to process sequential data that depends on time and order. RNNs have short-term memory, allowing them to understand the context of most recent events. While processing the sequential data, it considers a single time step at a time. This sequential processing helps in capturing short-term dependencies between adjacent time steps. However, for long-term sequential data dependencies, RNNs face a vanishing gradient problem, which causes the network's gradient to become very small during training. The BLSTM architecture has been introduced to address this issue and model long-term dependencies. However, the BLSTM architecture is more complex than the RNNs. If not properly regularized, BLSTM can be prone to overfitting, especially on a small dataset. Overfitting can lead to poor generalization performance. In this work, we achieve a combination of short-term and long-term data dependencies by employing a short-term memory neural network, an RNN, in conjunction with a long-term memory neural network using BLSTM. This combination helps to reduce the complexity of the BLSTM neural network. Figure 4 depicts the detailed neural network architecture.

RNN-BLSTM combines two learning algorithms: (1) real-time recurrent learning (RTRL), which is based

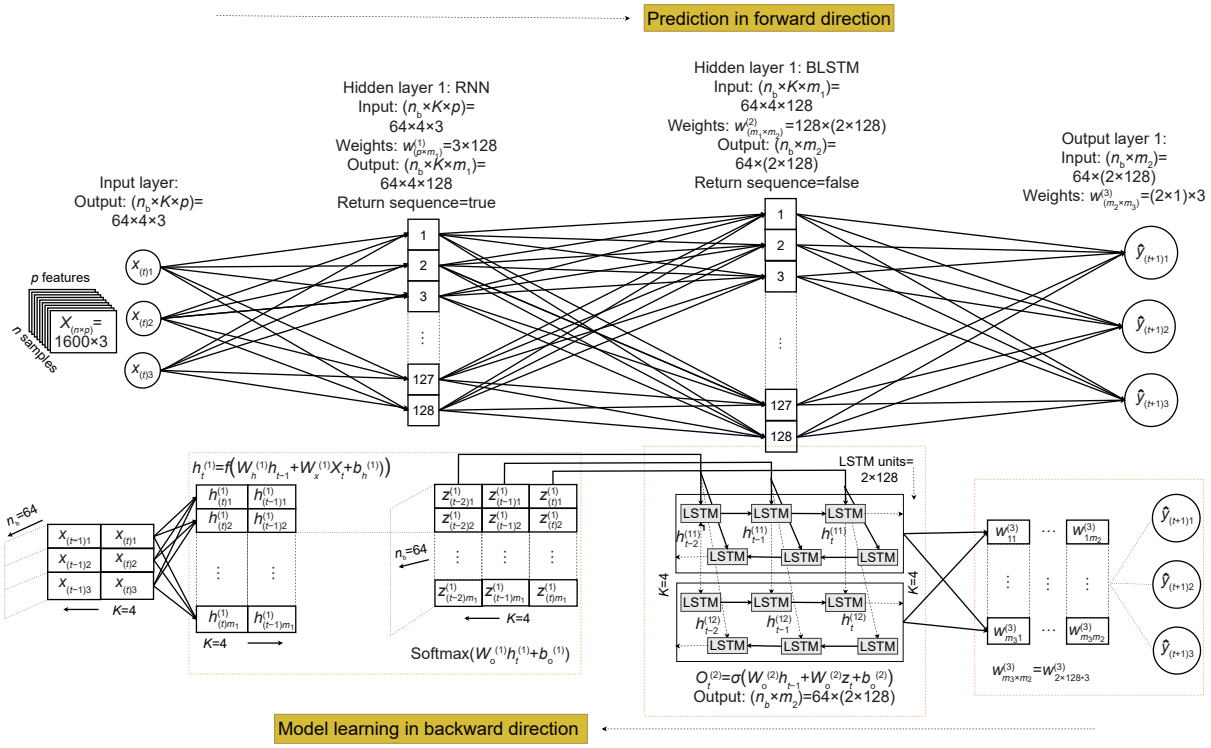



**Fig. 4 Schematic representation of the proposed neural network architecture, featuring a single-layered RNN with 128 hidden units and a single-layered BLSTM with 128 hidden units.**

on the gradient-based update rule; (2) BLSTM, which comprises two distinct hidden layers, one for forward and the other for backward learning, capturing past and future trends, respectively.

(1) Input layer: The input data samples encompass three traffic types (SMS, call, and Internet), each corresponding to one of the three input features.

(2) RNN: The single-layered RNN can be formulated as follows:

$$h_{(t-k)} = f(X_{(t-k)}, h_{(t-k-1)}), k \in \{0, 1, 2, \ldots, K-1\} \quad (2)$$

In this context, the function $f(\cdot)$ represents the recursive application of a transfer function to the input data window. At a given time $t$, the RNN takes an input sequence $X_t$ and produces the corresponding output $Z_t$. By examining the transition of the hidden state from $h_{(t-k-1)}$ to $h_{(t-k)}$ and making use of the Eq. (2), the recurrent state of the RNN can be formally described as

$$h_t = \tanh\left(W_h^{(1)} h_{(t-1)} + W_x^{(1)} X_t + b_h^{(1)}\right) \quad (3)$$

where the activation function, hyperbolic tangent function denoted as tanh, is defined as $\tanh(x) = \dfrac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$. Additionally, $W_h$ and $W_x$ are the weight vectors for the recurrent and input neurons, respectively. The previous state is represented as $h_{(t-1)}$, and $b_h$ denotes the bias vector at the current recurrent state. The superscript indicates the $i$-th layer within the RNN model. The output of the recurrent state at time $t$ is designated as

$$Z_t = \text{Softmax}\left(W_o^{(1)} h_{(t)} + b_o^1\right) \quad (4)$$

In the above equation, $W_o$ represents the weight vector at the output neuron, $h_t$ denotes the recurrent state and $b_o$ stands for the bias at the output neuron. Finally, $\text{Softmax} = \dfrac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$ is applied as the final activation function to normalize each output of the recurrent state to a probability distribution over the predicted target values.

(3) BLSTM neural network: The BLSTM neural network is characterized by its design, which incorporates three essential gates to regulate the flow of information:

• Forget gate: The forget gate controls what

information from the previous time step should be retained and what should be discarded. It is denoted as $f_t$.

• Input gate: The input gate decides which new information is essential to store in the cell state at the current time step. It is represented as $i_t$.

• Output gate: The output gate determines what part of the cell state is used for the current prediction. It is indicated as $o_t$.

At each time step $t$, the output of the RNN layer, denoted as $Z_t$, is comprised of elements $z_{(t)1}$, $z_{(t)2}$, and so on, up to $z_{(t)m_2}$. These outputs are then passed into the LSTM layer. Additionally, the hidden layer output is represented as $h_t$. The cell input state is $\tilde{C}_t$, and the output state is $C_t$. The gate states $f_t$, $i_t$, and $o_t$ have similar dimensions. Both $h_t$ and $C_t$ are propagated to the subsequent time step. Equation (5) introduces the mathematical representations of activation functions.

$$f_t = \sigma(W_{\mathrm{f}} \cdot (h_{t-1}, z_t) + b_{\mathrm{f}}) \tag{5}$$

where the symbol $\sigma$ denotes the sigmoid activation function, defined as $\frac{1}{1+\exp(-z)}$. The sigmoid function constrains prediction values to the range [0, 1], while the tanh function regulates values within the interval [−1, 1]. The BLSTM architecture relies on weight matrices $W_{\mathrm{f}}$, $W_{\mathrm{i}}$, $W_{\mathrm{o}}$, and $W_{\mathrm{c}}$, along with biases $b_{\mathrm{f}}$, $b_{\mathrm{i}}$, $b_{\mathrm{o}}$, and $b_{\mathrm{c}}$, for each gate and cell input.

The forget gate plays a crucial role in determining which information should be retained in the cell state within the BLSTM architecture. It operates within the range of 0 to 1, where a value of 0 signifies the data's irrelevance from the prior state. To achieve this, the forget gate combines the output of the previous state, denoted as $h_{t-1}$, with the current input, represented by $z_t$. Subsequently, these combined values are subjected to the sigmoid activation function, as expressed in Eq. (5).

The input gate takes the output of the previous state $h_{t-1}$ and the current input $z_t$ and then processes these values through both the sigmoid and tanh functions, as shown in Eqs. (6) and (7). The input gate determines how much the input value influences the memory cells.

$$i_t = \sigma(W_{\mathrm{i}} \cdot (h_{t-1}, z_t) + b_{\mathrm{i}}) \tag{6}$$

$$\tilde{C}_t = \tanh(W_{\mathrm{c}} \cdot (h_{t-1}, z_t) + b_{\mathrm{c}}) \tag{7}$$

At this point, the previous cell state $\tilde{C}_t$ is updated to the new cell state $C_t$ as shown in Eq. (8).

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{8}$$

where $\odot$ is Hadamard's (element-wise) product.

The output gate in BLSTM plays a pivotal role in determining the subsequent hidden state. It takes into account both the output of the previous state, $h_{t-1}$, and the current input, $z_t$. These values are processed through a sigmoid function, as exemplified in Eq. (9). Subsequently, the updated state, $C_t$, undergoes further refinement through the tanh function as demonstrated in Eq. (10). The final output is computed by multiplying the outcomes of the sigmoid and tanh functions.

$$O_t = \sigma(W_{\mathrm{o}} \cdot (h_{t-1}, z_t) + b_{\mathrm{o}}) \tag{9}$$

$$h_t = o_t \odot \tanh(C_t) \tag{10}$$

The new cell state and the new hidden unit output are then carried over to the next time step.

In the context of BLSTM, the input sequence is initially processed by the LSTM forward cell $\overrightarrow{O} = \mathrm{LSTM_f}(X_t)$, and subsequently by the LSTM backward cell $\overleftarrow{O} = \mathrm{LSTM_b}(X_t)$. The resulting output is a fusion of the outputs from these two cells, resulting in a combined output $O = [\overrightarrow{O}, \overleftarrow{O}]$. This bidirectional sequential data processing improves learning capabilities by considering information from past and future contexts, facilitating more efficient learning.

(4) Output dense layer: The return sequence is false in the LSTM layer shows that it will only return the last hidden state output. The LSTM layer is followed by the fully connected output dense layer, which gives the desired predicted value $\tilde{Y}_{t+1}$.

### 4.3 DSTL

The classical TL-based prediction model is employed for target gNBs exhibiting comparable traffic patterns. In contrast, a DSTL-based prediction model is applied to the target gNBs that demonstrate a lower correlation in traffic patterns compared to the source gNB. Figures 1 and 5 depicts the proposed DSTL technique.

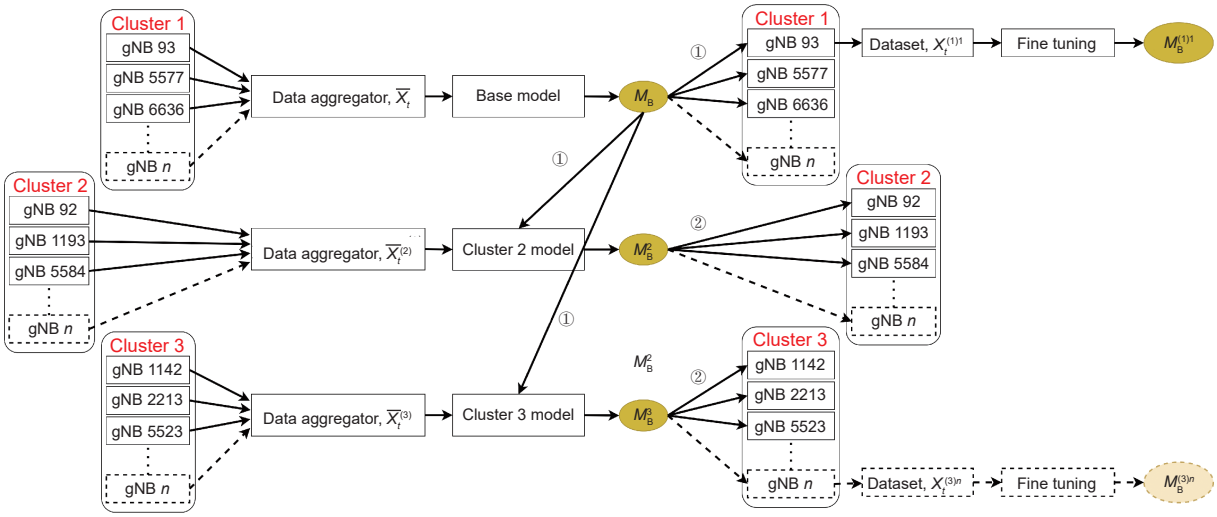



**Fig. 5 Representation of a DSTL-based prediction model training where ① denotes the first-step TL and ② denotes the second-step TL, conducted on three clusters and nine gNBs.**

(1) Initial base model: The initial step in the proposed framework involves acquiring an initial base model, which serves as a reference starting point for the gNBs. For this purpose, the data is aggregated within the cluster using the following equation:

$$\overline{X}_t^{(k)} = \frac{1}{n}\sum_{i-1}^{n} X_t^{(k)i} \tag{11}$$

In the above equation, $\overline{X}_t^{(k)}$ represents the aggregated data for time $t$ within a cluster $k$. It is calculated as the average of $n$ individual data samples received from $n$ gNBs denoted as $X_t^{(k)n}$. This aggregated data combines data patterns from all the gNBs within a specific cluster. Subsequently, the initial base model $M_B$ is trained using the proposed RNN-BLSTM prediction model discussed in Section 4.2. This initial model is analogous to the initial guess of the hyperparameters. For instance, the hyperparameters may include values such as a sequence length of 4, a batch size of 64, a learning rate of 0.01, a single RNN layer with 128 units, and a single BLSTM layer with 128 units.

(2) First-step TL: As previously mentioned, the initial base model, denoted as $M_B$, is trained on the aggregated dataset within Cluster 1. Cluster 2 is comprised of gNBs 92, 1193, and 5584, whereas the Cluster 3 consists of gNBs 1142, 2213, and 5523. The method of ITL is employed to facilitate the transfer of the base model from the source domain, represented by Cluster 1, to the target domain in Cluster 2, specifically gNB 5584. However, Table 1 illustrates that the traffic pattern of gNB 5584 exhibits only a 24% correlation with the traffic pattern of Cluster 1. This low correlation necessitates an extensive fine-tuning process for the transferred base model $M_B$ to achieve optimal performance in the target domain, which results in higher computational demands and transferred model fine-tuning time, leading to increased computational requirements within Cluster 2 for every gNB.

For this purpose, to fully exploit the benefits of ITL,

**Table 1 Correlation of gNode 5584 with aggregated data of Clusters 1, 2, and 3, respectively.**

| Traffic type and average | Cluster 1 | | Cluster 2 | | Cluster 3 | |
|---|---|---|---|---|---|---|
| | Unsmoothed | Smoothed | Unsmoothed | Smoothed | Unsmoothed | Smoothed |
| SMS | 0.20 | 0.23 | 0.89 | 0.95 | 0.19 | 0.24 |
| Call | 0.28 | 0.29 | 0.90 | 0.95 | 0.14 | 0.15 |
| Internet | 0.24 | 0.25 | 0.93 | 0.96 | 0.33 | 0.34 |
| Average | 0.24 | 0.26 | 0.91 | 0.95 | 0.22 | 0.24 |

the first step, TL, involves fine-tuning the pre-trained base model on the traffic patterns of each gNB within the same cluster and on the aggregated traffic patterns of the other clusters.

(3) Second-step TL: The analysis of the cellular traffic dataset reveals that Cluster 1 shares an aggregated traffic pattern similarity of 48% with Cluster 2, while gNB 5584 exhibits a traffic pattern correlation of 95% with Cluster 2. This observation indicates that training a model on the aggregated dataset of Cluster 2 would confer simultaneous benefits to all gNBs within the same cluster. This approach effectively reduces the computational demands and model training time for every gNB in a cluster.

Following fine-tuning the transferred base model on the aggregated traffic patterns, Cluster 2 initiates a second-step ITL process to provide the cluster model $M_{\mathrm{B}}^{2}$ to each gNB within the Cluster 2. This same methodology can be extended to the other clusters within the network. The results demonstrate that the proposed DSTL technique achieves superior prediction accuracy compared to previous studies, even without any fine-tuning on the local dataset of each gNB. This DSTL technique is attributed to computational resource savings and the reduction in model training time, as it eliminates the need to train the model from scratch for each gNB. Additionally, this technique will alleviate the burden on core links, cloud servers, and gNBs, as the model is trained on the edge servers within the clusters. These optimizations contribute to more efficient and resource-friendly operations in the network.

## 5 Experimental setup

This section provides a comprehensive overview of the dataset collection process for clustering, as well as the hyperparameters used in the prediction model and the performance evaluation metrics employed in the study.

### 5.1 Dataset collection

The cellular coverage of Milan city is divided into 10 000 cells, as reported in Ref. [31]. For the purpose of our simulations, we focus on a subset of 1000 cells and utilize a dataset covering the initial 16 days from November 1st, 2013, to November 16th, 2013. The selected cells are strategically positioned within a $100\times100$ grid in a diagonal pattern. These cells exhibit sufficient required heterogeneity in their traffic patterns, which enables the creation of distinct clusters. In the case of the DSTL-based prediction model, 75% of the data is allocated for training, while the remaining 25% is reserved for testing and validation.

### 5.2 Prediction model hyperparameter

(1) Input layer: The dataset comprises 1600 data samples, encompassing three traffic types (SMS, call, and Internet), each corresponding to one of the three input features, $p=3$. During the training of the initial base model, a batch size of $64\leqslant n_b\leqslant128$ with a step size of 64 is employed to strike a balance between accurate gradient estimation and computational cost. Moreover, a timestamp $1\leqslant K\leqslant10$ with a step size of 1 is used in the training process of the base model.

(2) First hidden layer: In the proposed DSTL-based prediction model, a single-layered RNN is utilized, where hidden units are denoted as $m_1$. The input size of the RNN layer is $n_b\times K\times p$. Consequently, this RNN layer produces an output with dimensions of $n_b\times K\times m_1$, where $16\leqslant m_1\leqslant256$ with a step size of 16.

(3) Second hidden layer: BLSTM layer processes input data of dimensions $n_b\times K\times m_1$ obtained from the preceding RNN layer. Subsequently, it generates an output of dimensions $n_b\times K\times(2\times m_2)$. During the hyperparameter grid search process, $m_2$ is tuned to achieve optimal performance and is expected to take on a similar value as the RNN layer. Given that BLSTM inherently possesses the capability to manage sequential data in long-term memory, the choice of timestamp for the BLSTM layer aligns with that of the RNN layer.

(4) Output layer: The BLSTM layer is followed by the dense output layer of shape $(2\times m_2)\times p$, which gives the desired predicted value $\tilde{Y}_{t+1}$.

(5) Model training parameters: During the model training process, we conducted training for a total of 100 epochs, employing a learning rate of $L_{\mathrm{R}}$ = 0.1, 0.01, and 0.001 for the base model.

(6) Hyperparameter grid search: In the process of

configuring hyperparameter combinations for the base model, a comprehensive set of 960 unique combinations was established. The objective is to apply all these combinations to the given dataset and identify the hyperparameters that yield the most optimal performance on the validation dataset.

### 5.3 Performance evaluation metric

This paper employs four performance metrics for evaluating the proposed DSTL-based prediction model. These metrics involve a comparison between the actual values $y_i$ and the predicted values $\hat{y}_i$ of the time series. The following four metrics are included:

(1) Mean absolute error (MAE): This metric assesses the average inaccuracy expected in the predicted time series. A smaller MAE indicates that the predicted values closely resemble the actual values.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{12}$$

(2) Mean absolute percentage error (MAPE): This metric calculates the percentage of the average absolute difference between the actual and predicted values relative to the actual values. It is suitable for datasets without zeros or extreme values.

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\% \tag{13}$$

(3) Mean squared error (MSE): This metric calculates the average of the squared differences between the actual and predicted values. It is susceptible to outliers or extreme errors, as the squared terms amplify their impact on the overall error measurement.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{14}$$

(4) Root mean squared error (RMSE): This metric is defined as the square root of the MAE. RMSE is useful for comparing with MAE to evaluate the size of errors in the time series forecast. A larger gap between MAE and RMSE indicates a more erratic error size. Similar to MSE, RMSE also penalizes outliers or extreme errors more than small errors.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{15}$$

## 6 Performance evaluation

In this section, we present the results of the experiments conducted on the real-world dataset, as described in Section 5.1.

### 6.1 Initial base model

For the simulation, we selecte Cluster 1 to train the base model using the RNN-BLSTM neural network, as detailed in Section 4.2. We utilize a well-established grid search method to identify the most suitable hyperparameters from a predefined list of combinations, as outlined in Section 5.2. The resulting model hyperparameters are detailed in Table 2 and are used to label the hyperparameters in Fig. 1. The base model is derived from a total of 960 unique hyperparameter combinations.

Figure 6 displays the training and validation loss, which is computed using MSE as the loss function, over 100 epochs. The curves for both training and validation are closely aligned, indicating a well-fitted prediction model. The model hyperparameters have varying impacts on model performance, as illustrated

**Table 2 Summary of the RNN-BLSTM model.**

| Layer (type) | Output shape | Number of paramters |
|---|---|---|
| Simple_rnn (SimpleRNN) | (none, 4, 128) | 39 424 |
| Dropout (dropout) | (none, 4, 128) | 0 |
| Bidirectional (bidirectional) | (none, 256) | 337 920 |
| Dropout (dropout) | (none, 256) | 0 |
| Dense (dense) | (none, 3) | 387 |

Note: The numbers of total parameters, trainable parameters, and non-trainable parameters are 394 627, 377 731, and 16 896, respectively.

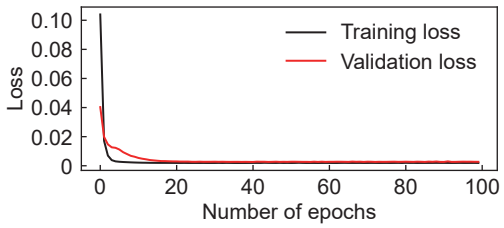



**Fig. 6 Training and validation loss vs. epochs.**

in Fig. 7, which shows the performance of model hyperparameters in terms of MSE. It is evident that the MSE is minimized when using $L_R = 0.01$, $K = 4$, and 128 hidden units in both the RNN and BLSTM layers, as depicted in Figs. 7a−7d, respectively.

The evaluation of the proposed RNN-BLSTM prediction model for multivariate spatio-temporal cellular traffic is depicted in Fig. 8. Specifically, Figs. 8a, 8c, and 8e present the comparisons between the actual and predicted SMS, call, and Internet activities within the cellular traffic, respectively. This visual representation illustrates the model's performance in capturing the multivariate spatio-temporal time series dynamics. In contrast, Figs. 8b, 8d, and 8f display the per-sample absolute errors for SMS, call, and Internet traffic predictions. The spikes in the error graph are due to the abrupt changes in cellular traffic patterns, and it can be analyzed that the trained model predicts and adapts to these changes. Furthermore, Table 3 provides the performance evaluation metrics for the base model. MAPE values are 9.64, 10.78, and 3.11 for SMS, call, and Internet traffic, respectively.

### 6.2 First-step transfer learning

Table 3 presents the results of the first-step TL technique, which is analogous to traditional TL methods. As indicated in Figs. 1 and 4, Cluster 1 encompasses gNBs 93, 5577, and 6636. Due to the similarity in traffic patterns between the aggregated dataset of Cluster 1 and gNB 93, the first-step TL is effective for gNB 93, resulting in a MAPE of just 4.67%. However, gNBs 5584 and 1142 belong to Clusters 2 and 3, respectively, and exhibit different traffic patterns compared to Cluster 1. Consequently, the MAPE for Internet traffic is 9.52% for gNB 5584 and 12.10% for gNB 1142, indicating the need for substantial fine-tuning of the transferred model at each gNB. Additional performance metrics are also available in Table 3.

Instead of directly providing the model to the gNBs in other clusters, the approach involves initially

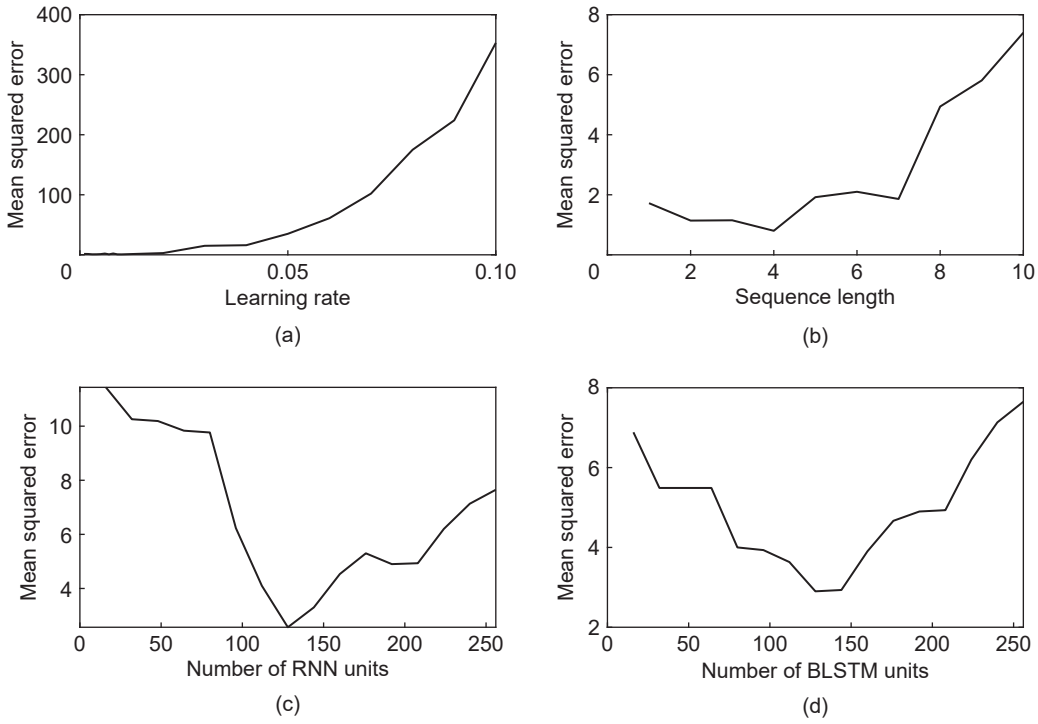



**Fig. 7 Performance of the base RNN-BLSTM prediction model in terms of MSE on different hyperparameters such as (a) learning rate, (b) sequence length, (c) number of RNN units, and (d) number of BLSTM units.**

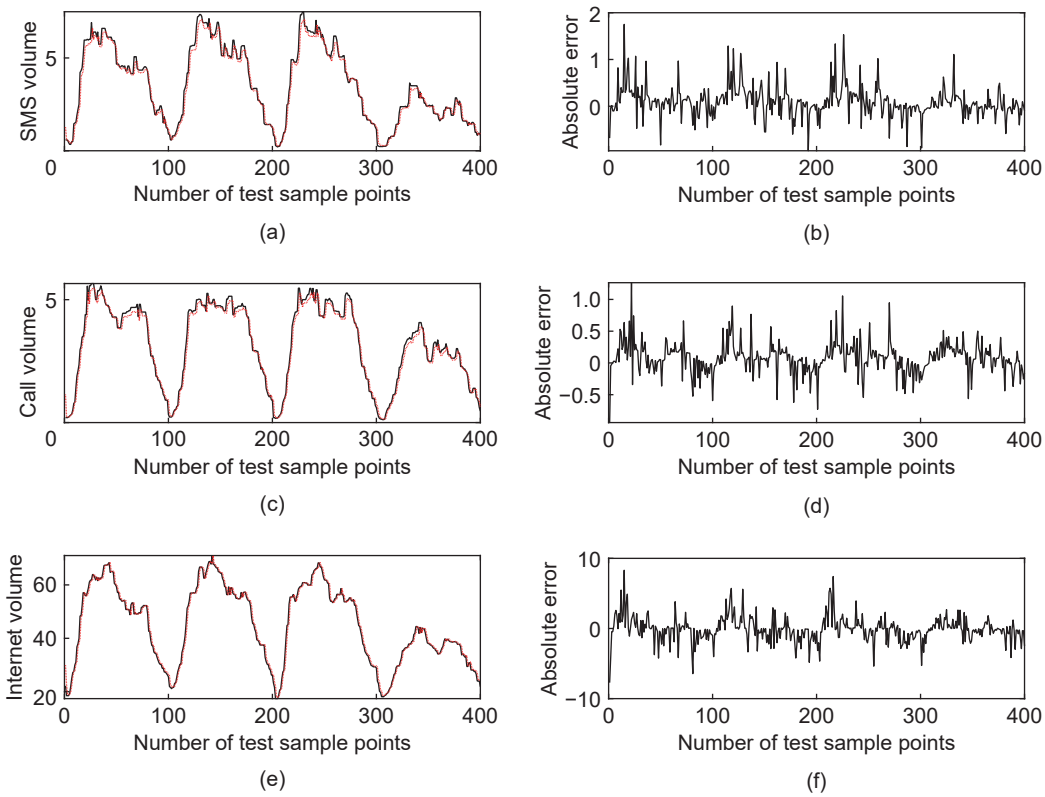



**Fig. 8 Prediction performance of base model on SMS, call, and Internet traffic using the aggregated data of Cluster 1. Specifically, (a), (c), and (e) show the actual and predicted traffic volumes for SMS, call, and Internet traffic, i.e., present the comparisons between the actual and predicted SMS, call, and Internet activities within the cellular traffic, respectively. In contrast, (b), (d), and (f) display the per-sample absolute errors for SMS, call, and Internet traffic predictions. In (a), (c), and (e), the black solid line represents the actual, while the red dotted line represents the predicted cellular activities.**

**Table 3 Performance metrics of the base model (Cluster 1) when transferred to gNodes 93 (Cluster 1), 5584 (Cluster 2), and 1142 (Cluster 3) using traditional TL approach.**

| Cellular traffic type | Performance evaluation metrics | Base model (Cluster 1) | Transferred model | | |
|---|---|---|---|---|---|
| | | | gNB 93 (Cluster 1) | gNB 5584 (Cluster 2) | gNB 1142 (Cluster 3) |
| SMS | MAE | 0.230 | 0.051 | 0.130 | 0.043 |
| | MAPE (%) | 9.64 | 14.40 | 27.21 | 23.80 |
| | MSE | 0.120 | 0.006 | 0.035 | 0.004 |
| | RMSE | 0.350 | 0.075 | 0.190 | 0.062 |
| Call | MAE | 0.190 | 0.060 | 0.110 | 0.031 |
| | MAPE (%) | 10.78 | 12.30 | 21.73 | 29.35 |
| | MSE | 0.070 | 0.008 | 0.020 | 0.003 |
| | RMSE | 0.260 | 0.089 | 0.150 | 0.049 |
| Internet | MAE | 1.200 | 0.540 | 1.410 | 0.420 |
| | MAPE (%) | 3.11 | 4.67 | 9.52 | 12.10 |
| | MSE | 3.380 | 0.710 | 3.070 | 0.890 |
| | RMSE | 1.840 | 0.840 | 1.750 | 0.950 |

training the model on the aggregated dataset of each cluster. This technique enables collective fine-tuning of the model for all the gNBs within a particular cluster, resulting in computational resource savings and reduced model fine-tuning time for all the gNBs within that cluster. Furthermore, based on the insights gained from the base model training, the hyperparameter combinations are reduced by narrowing the range of

hyperparameter values closer to the optimal parameters obtained from the base model, as shown in Table 4. Specifically, the range of sequence length $K$ is reduced to only there values from the initial ten. This fine-tuning process enhances the model's performance, resulting in a reduction in the MAPE to 4.30% for Cluster 2 and 3.58% for Cluster 3, as shown in Table 4. At this point, the prediction model is ready for evaluation across all gNBs in the given scenario, with a particular focus on Clusters 2 and 3, where the results of the proposed DSTL technique are assessed.

### 6.3 Second-step transfer learning

In the second-step TL phase, the model trained on the aggregated data of Clusters 2 and 3 is transferred to all participating gNBs within their respective clusters. To assess the benefits of the proposed DSTL technique, gNB 5584 from Cluster 2 is selected for simulation. Figure 9 displays the performance evaluation metrics for gNB 5584 when utilizing the pre-trained prediction model received from Clusters 1 and 2. Table 1 reveals that gNB 5584 exhibits a weak correlation of 0.24 with Cluster 1 and a stronger correlation of 0.91 with Cluster 2. Due to higher pattern similarity between the aggregated dataset of Cluster 2 and gNB 5584, the model trained on aggregated dataset Cluster 2 delivers superior performance on gNB 5584.

The performance evaluation metrics presented in Table 4 demonstrate that the model trained on the aggregated dataset of Cluster 2 attains a MAPE of 9.73%, 9.85%, and 2.97% for SMS, call, and Internet traffic, respectively, on gNB 5584 without requiring any additional fine-tuning. These results underscore the effectiveness of the DSTL technique for training prediction models across various gNBs, with reduced computational resources and model training time.

Figure 10 illustrates the advantages of the proposed DSTL technique, emphasizing the reductions in hyperparameter combinations, model training time, memory utilization, and processor requirements. The DSTL technique significantly decreases the hyperparameter combinations from 960 to 192 for each gNB, as shown in Fig. 10a, by narrowing the hyperparameter range closer to the pre-trained initial base model. This reduction in hyperparameter combinations subsequently results in a four-fold reduction in the model execution time, as shown in

**Table 4 Illustration of the proposed DSTL-based prediction model training, specifying the model hyperparameter (HP) range and presenting performance metrics.**

| Model training phase | Cluster | HP range and step size | Optimal HP | Performance | | |
|---|---|---|---|---|---|---|
| | | | | SMS | Call | Internet |
| Base model training | Cluster 1 | $1 \leqslant K \leqslant 10$, 1<br>$64 \leqslant n_b \leqslant 128$, 64<br>$L_R = 0.1, 0.01, 0.001$<br>$16 \leqslant m_1 \leqslant 256, 16 \leqslant m_2 \leqslant 256$, step size = 16 | $K = 4$,<br>$n_b = 64$,<br>$L_R = 0.01$,<br>$m_1 = 128$, $m_2 = 128$ | MAE = 0.210,<br>MAPE = 9.22%,<br>MSE = 0.0950,<br>RMSE = 0.310 | MAE = 0.150,<br>MAPE = 8.78%,<br>MSE = 0.0510,<br>RMSE = 0.220 | MAE = 1.100,<br>MAPE = 2.73%,<br>MSE = 2.260,<br>RMSE = 1.500 |
| First-step TL | Cluster 2 | $3 \leqslant K \leqslant 5$, 1<br>$64 \leqslant n_b \leqslant 128$, 64<br>$L_R = 0.01, 0.001$<br>$16 \leqslant m_1 \leqslant 256, 16 \leqslant m_2 \leqslant 256$, step size = 16 | $K = 5$,<br>$n_b = 64$,<br>$L_R = 0.001$,<br>$m_1 = 256$, $m_2 = 256$ | MAE = 0.0440,<br>MAPE = 10.77%,<br>MSE = 0.005,<br>RMSE = 0.070 | MAE = 0.0420,<br>MAPE = 10.08%,<br>MSE = 0.004,<br>RMSE = 0.058 | MAE = 0.420,<br>MAPE = 4.30%,<br>MSE = 0.350,<br>RMSE = 0.590 |
| | Cluster 3 | $3 \leqslant K \leqslant 5$, 1<br>$64 \leqslant n_b \leqslant 128$, 64<br>$L_R = 0.01, 0.001$<br>$16 \leqslant m_1 \leqslant 256, 16 \leqslant m_2 \leqslant 256$, step size = 16 | $K = 4$,<br>$n_b = 128$,<br>$L_R = 0.001$,<br>$m_1 = 64$, $m_2 = 64$ | MAE = 0.088,<br>MAPE = 10.82%,<br>MSE = 0.016,<br>RMSE = 0.130 | MAE = 0.082,<br>MAPE = 15.09%,<br>MSE = 0.016,<br>RMSE = 0.130 | MAE = 0.830,<br>MAPE = 3.58%,<br>MSE = 2.120,<br>RMSE = 1.460 |
| Second-step TL | gNB 5584 (Cluster 2) | No further hyperporameter eyploration | $K = 5$,<br>$n_b = 64$,<br>$L_R = 0.001$,<br>$m_1 = 256$, $m_2 = 256$ | MAE = 0.210,<br>MAPE = 9.73%,<br>MSE = 0.096,<br>RMSE = 0.310 | MAE = 0.170,<br>MAPE = 9.85%,<br>MSE = 0.054,<br>MSE = 0.230 | MAE = 1.240,<br>MAPE = 2.97%,<br>MSE = 2.740,<br>RMSE = 1.660 |

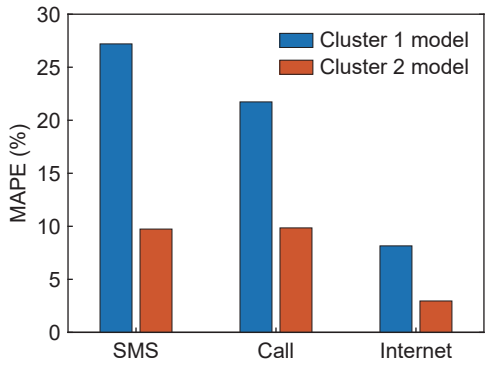



**Fig. 9 MAPE for different traffic types (SMS, call, and Internet) at gNB 5584, using the pre-trained model received from Clusters 1 and 2.**

Fig. 10b. Memory usage during model training is reduced to 42% from the initial 76%, while the processor requirements remain largely unchanged. This demonstrates the efficiency and resource savings achieved with the DSTL technique.

### 6.4 Comparison of DSTL-based RNN-BLSTM

The prediction accuracy of the proposed DSTL-based RNN-BLSTM prediction model is compared with statistical and machine learning-based prediction models, including SVR with a radial basis function, ARIMA(4,1,0), a single-layered RNN, and a single-layered BLSTM with 256 units. Grid search is employed to optimize the performance of all the prediction models. As illustrated in Fig. 11, the proposed DSTL-based RNN-BLSTM model exhibits superior predictive capabilities compared to the other prediction models on gNB 5584, as evidenced by its lower MAE, MAPE, MSE, and RMSE values.

Table 5 provides a comprehensive comparison of the proposed DSTL technique with existing literature[53−60] in the field of cellular traffic prediction. The DSTL-based RNN-BLSTM prediction model demonstrates superior performance compared to other frameworks, especially in predicting both univariate and multivariate cellular traffic patterns. For comparison, we utilize vital metrics such as MAPE and RMSE to evaluate and highlight the achieved results.

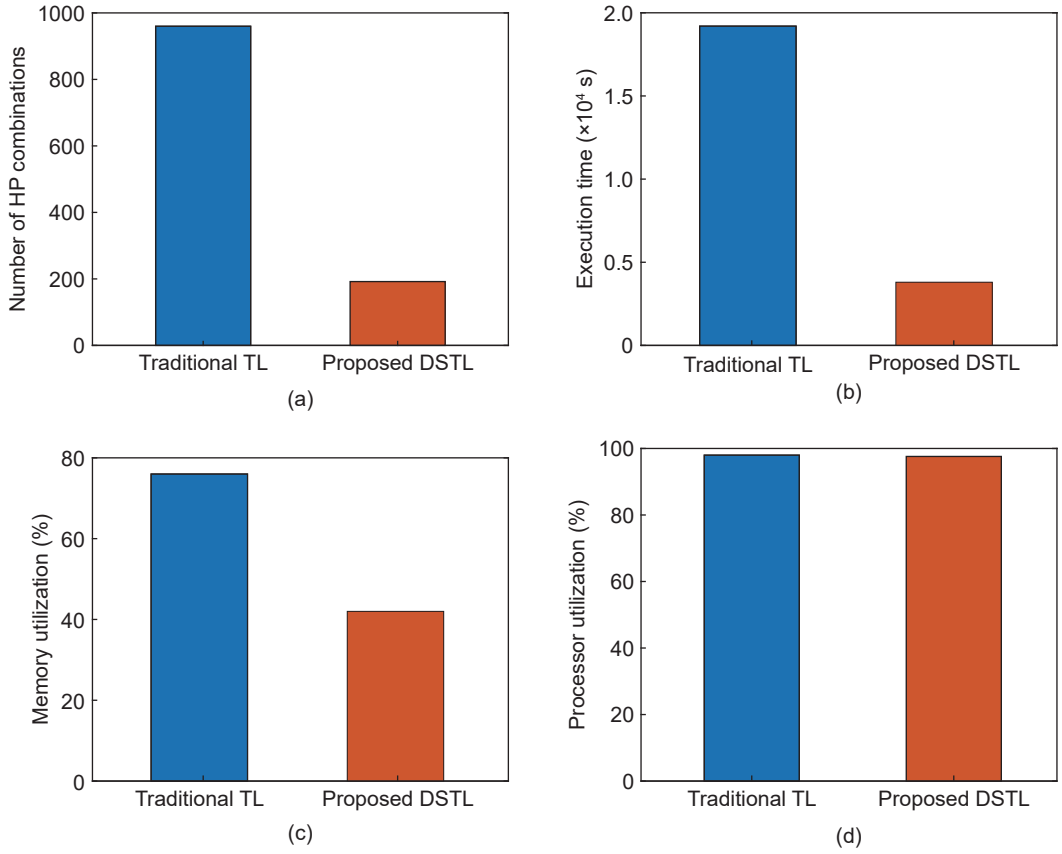



**Fig. 10 Benefits of the proposed DSTL approach on (a) number of HP combinations, (b) prediction model training time, (c) memory utilization during model training, and (d) processor utilization.**

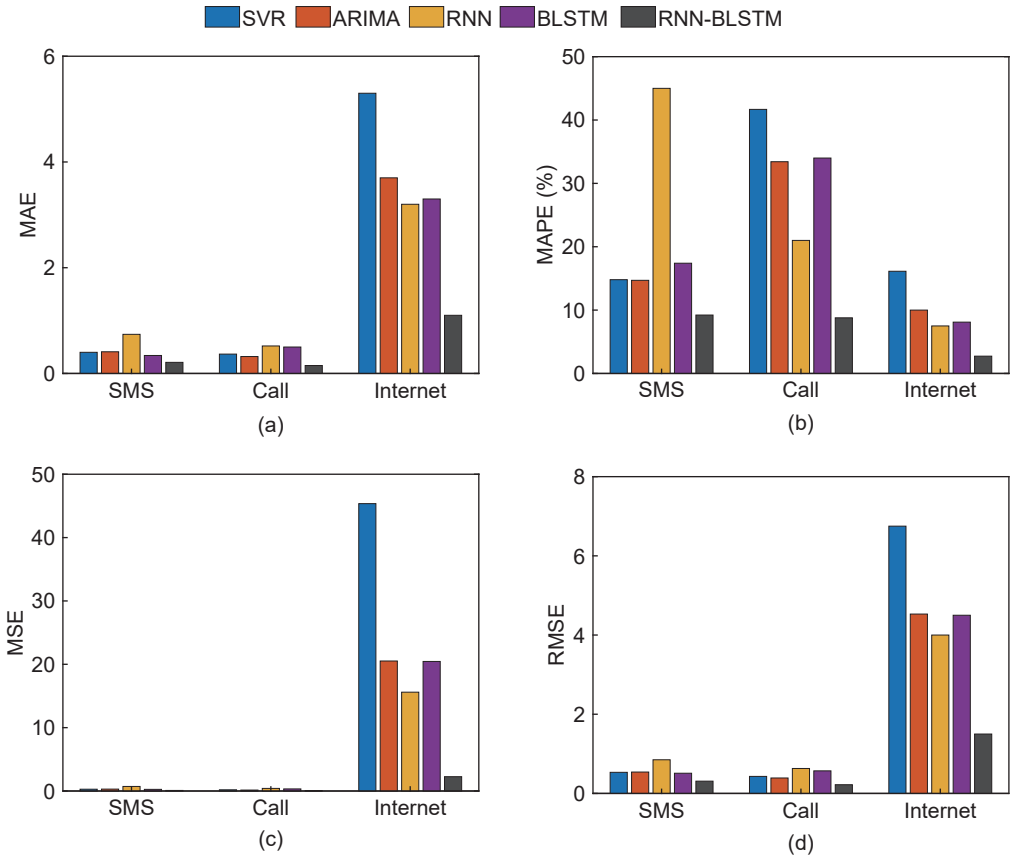



**Fig. 11 Comparison of the proposed DSTL-based RNN-BLSTM prediction model with other existing prediction models.**

**Table 5 Comparison of the proposed DSTL-based prediction model with existing literature.**

| Model | Evaluation metric | Univariate dataset | Multivariate dataset | Result |
|---|---|---|---|---|
| Model in Ref. [53] | MAPE | √ | — | 29.46% |
| Model in Ref. [54] | MAPE | √ | — | 9.13% |
| Model in Ref. [55] | RMSE | √ | — | 0.620 |
| Model in Ref. [56] | MAPE | — | √ | 13.40% |
| Model in Ref. [57] | MAPE | — | √ | 11.00% |
| Model in Ref. [58] | MAPE | — | √ | 13.00% |
| Model in Ref. [59] | MAPE | — | √ | 16.00% |
| Model in Ref. [60] | MAPE | — | √ | 18.00% |
| DSTL | MAPE | √ | — | MAPE on SMS = 9.73%<br>MAPE on call = 9.85%<br>MAPE on Internet = 2.97% |
| DSTL | MAPE | — | √ | 7.52% |
| DSTL | RMSE | √ | — | RMSE on SMS = 0.310<br>RMSE on call = 0.230<br>RMSE on Internet = 1.660 |
| DSTL | RMSE | — | √ | 0.730 |

Note: √ shows which type of dataset is utilized and — shows that the dataset is not been used.

## 7 Conclusion

This study explore the effectiveness of the proposed DSTL-based prediction model for anticipating heterogeneous multivariate spatio-temporal cellular traffic patterns. Our approach involve segmenting the cellular coverage area into distinct clusters based on

traffic distribution correlations. Using the RNN-BLSTM neural network, we develop a base model trained on the aggregated dataset of a designated base cluster. Employing the inductive TL approach, in the first-step TL, we transfer this model to other clusters, fine-tuning it on the aggregated datasets of respective clusters.

In the second-step TL, the intra-cluster model is derived and deployed to individual gNBs within each cluster. The DSTL technique alleviates the need for exhaustive hyperparameter exploration during ML model training for each gNB. Our proposed approach achievs a low mean absolute percentage error of 2.97%, 9.85%, and 9.73% in predicting spatio-temporal Internet, call, and SMS traffic, respectively, while optimizing time, data, and computational resource utilization.

## Acknowledgment

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme (No. 739578); the ADROIT6G project of the Smart Networks and Services Joint Undertaking (No. 101095363); the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

## Conflict of Interest

The authors declare no conflict of interest.

## References

[1] M. K. Shehzad, L. Rose, M. M. Butt, I. Z. Kovács, M. Assaad, and M. Guizani, Artificial intelligence for 6G networks: Technology advancement and standardization, *IEEE Veh. Technol. Mag.*, vol. 17, no. 3, pp. 16–25, 2022.

[2] P. Rohini, S. Tripathi, C. M. Preeti, A. Renuka, J. L. A. Gonzales, and D. Gangodkar, A study on the adoption of wireless communication in big data analytics using neural networks and deep learning, in *Proc. 2nd Int. Conf. Advance Computing and Innovative Technologies in Engineering* (*ICACITE*), Greater Noida, India, 2022, pp. 1071–1076.

[3] S. Ashtari, I. Zhou, M. Abolhasan, N. Shariati, J. Lipman, and W. Ni, Knowledge-defined networking: Applications, challenges and future work, *Array*, vol. 14, p. 100136, 2022.

[4] E. Coronado, R. Behravesh, T. Subramanya, A. Fernàndez-Fernàndez, S. Siddiqui, X. Costa-Pérez, and R. Riggio, Zero touch management: A survey of network automation solutions for 5G and 6G networks, *Surv. Tutor.*, vol. 24, no. 4, pp. 2535–2578, 2022.

[5] K. Mehmood, K. Kralevska, and D. Palma, Intent-driven autonomous network and service management in future cellular networks: A structured literature review, *Comput. Netw.*, vol. 220, p. 109477, 2023.

[6] Z. Wang, J. Hu, G. Min, Z. Zhao, Z. Chang, and Z. Wang, Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach, *Comput. Netw.*, vol. 19, no. 4, pp. 5722–5731, 2023.

[7] E. S. Escriche, S. Vassaki, and G. Peters, A comparative study of cellular traffic prediction mechanisms, *Wireless Network.*, vol. 29, no. 5, pp. pp.2371–2389, 2023.

[8] S. Tanberk and O. Demir, Multivariate modeling and analysis for cellular traffic prediction using call detail records, in *Proc. 7th Int. Conf. Computer Science and Engineering* (*UBMK*), Diyarbakir, Türkiye, 2022, pp. 311–316.

[9] W. Jiang, Cellular traffic prediction with machine learning: A survey, *Expert Syst. Appl.*, vol. 201, pp. 117163, 2022.

[10] A. Azari, P. Papapetrou, S. Denic, and G. Peters, Cellular traffic prediction and classification: A comparative evaluation of LSTM and ARIMA, in *Proc. 22nd Int. Conf. Discovery Science*, Split, Croatia, pp. 129–144.

[11] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, Wireless traffic prediction with scalable Gaussian process: Framework, algorithms, and verification, *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291–1306, 2019.

[12] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai, and H. Zhang, The learning and prediction of application-level traffic data in cellular networks, *IEEE Trans. Wirel. Commun.*, vol. 16, no. 6, pp. 3899–3912, 2017.

[13] Y. Shu, M. Yu, O. Yang, J. Liu, and H. Feng, Wireless traffic modeling and prediction using seasonal ARIMA models, *IEICE Trans. Commun.*, vol. 88, no. 10, pp. 3992–3999, 2005.

[14] B. Zhou, D. He, and Z. Sun, Traffic predictability based on ARIMA/GARCH model, in *Proc. 2nd Conf. Next Generation Internet Design and Engineering*, Valencia, Spain, 2006, p. 8.

[15] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, Big data driven mobile traffic understanding and forecasting: A time series approach, *IEEE Trans. Serv. Comput.*, vol. 9, no. 5, pp. 796–805, 2016.

[16] X. Chen, Y. Jin, S. Qiang, W. Hu, and K. Jiang, Analyzing

and modeling spatio-temporal dependence of cellular traffic at city scale, in *Proc. 2015 IEEE Int. Conf. Communications* (*ICC*), London, UK, 2015, pp. 3585–3591.

[17] R. Li, Z. Zhao, X. Zhou, J. Palicot, and H. Zhang, The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice, *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 234–240, 2014.

[18] I. Lohrasbinasab, A. Shahraki, A. Taherkordi, and A. Delia Jurcut, From statistical-to machine learning-based network traffic prediction, *Trans. Emerg. Telecommun.*, vol. 33, no. 4, p. e4394, 2022.

[19] M. Mohseni, S. Nikan, and A. Shami, Ai-based traffic forecasting in 5G network, in *Proc. 2022 IEEE Canadian Conf. Electrical and Computer Engineering* (*CCECE*), Halifax, NS, Canada, IEEE, 2022, pp. 188–192.

[20] A. R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model, *Inf. Sci.*, vol. 484, pp. 302–337, 2019.

[21] H. Sun, H. X. Liu, H. Xiao, and B. Ran, Short term traffic forecasting using the local linear regression model, https://escholarship.org/uc/item/540301xx, 2002.

[22] N. I. Sapankevych and R. Sankar, Time series prediction using support vector machines: A survey, *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, 2009.

[23] S. Jaffry and S. F. Hasan, Cellular traffic prediction using recurrent neural networks, in *Proc. 5th Int. Symp. Telecommunication Technologies* (*ISTT*), Shah Alam, Malaysia, 2020, pp. 94–98.

[24] W. A. Aziz, H. K. Qureshi, A. Iqbal, and M. Lestas, Accurate prediction of streaming video traffic in TCP/IP networks using dpi and deep learning, in *Proc. 2020 Int. Wireless Communications and Mobile Computing* (*IWCMC*), Limassol, Cyprus, 2020, pp. 310–315.

[25] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q. V. Pham, D. Niyato, E. Dutkiewicz, and W. J. Hwang, Transfer learning for future wireless networks: A comprehensive survey, arXiv preprint arXiv: 2102.07572, 2021.

[26] L. Zhang, C. Zhang, and B. Shihada, Efficient wireless traffic prediction at the edge: A federated meta-learning approach, *IEEE Commun. Lett.*, vol. 26, no. 7, pp. 1573–1577, 2022.

[27] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q. V. Pham, D. Niyato, E. Dutkiewicz, and W. J. Hwang, Transfer learning for wireless networks: A comprehensive survey, *Proc. IEEE*, vol. 110, no. 8, pp. 1073–1115, 2022.

[28] M. Iman, H. R. Arabnia, and K. Rasheed, A review of deep transfer learning and recent advancements, *Technologies*, vol. 11, no. 2, p. 40, 2023.

[29] K. R. Weiss and T. M. Khoshgoftaar, Comparing transfer learning and traditional learning under domain class imbalance, in *Proc. 16th IEEE Int. Conf. Machine Learning and Applications* (*ICMLA*), Cancun, Mexico, 2017, pp. 337–343.

[30] W. A. Aziz, I. Ioannou, M. Lestas, and V. Vassiliou, Dual-step transfer learning-based prediction model for next-generation intelligent cellular networks, in *Proc. 2023 IEEE Int. Conf. Communications Workshops* (*ICC Workshops*), Rome, Italy, 2023, pp. 260–265.

[31] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, A multi-source dataset of urban life in the city of Milan and the province of Trentino, *Sci. Data*, vol. 2, no. 1, p. 150055, 2015.

[32] C. Zhang, S. Dang, B. Shihada, and M. S. Alouini, Dual attention-based federated learning for wireless traffic prediction, in *Proc. IEEE INFOCOM 2021-IEEE Conf. Computer Communications*, Vancouver, Canada, 2021, pp. 1–10.

[33] D. Tikunov and T. Nishimura, Traffic prediction for mobile network using holt-winter's exponential smoothing, in *Proc. 15th Int. Conf. Software, Telecommunications and Computer Networks*, Split, Croatia, 2007, pp. 1–5.

[34] S. Nyaramneni, M. A. Saifulla, and S. S. Mehra, Internet traffic prediction in SDN using RF and XGB, in *Proc. Int. Conf. Computational and Bio Engineering*, Tirupati, India, 2020, pp. 153–159.

[35] G. Choudhury, D. Lynch, G. Thakur, and S. Tse, Two use cases of machine learning for SDN-enabled IP/optical networks: Traffic matrix prediction and optical path performance prediction [Invited], *J. Opt. Commun. Networking*, vol. 10, no. 10, pp. D52–D62, 2018.

[36] Y. Zhang, N. Han, T. Zhu, J. Zhang, M. Ye, S. Dou, and Z. Guo, Prophet: Traffic engineering-centric traffic matrix prediction, *IEEE ACM Trans. Netw.*, vol. 32, no. 1, pp. 822–832, 2024.

[37] H. H. Tong, C. R. Li, and J. R. He, Boosting feed-forward neural network for internet traffic prediction, in *Proc. 2004 Int. Conf. Machine Learning and Cybernetics*, Shanghai, China, 2004, pp. 3129–3134.

[38] D. Andreoletti, S. Troia, F. Musumeci, S. Giordano, G. Maier, and M. Tornatore, Network traffic prediction based on diffusion convolutional recurrent neural networks, in

*Proc. IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops* (*INFOCOM WKSHPS*), Paris, France, 2019, pp. 246–251.

[39] W. Jiang, Y. Zhang, H. Han, Z. Huang, Q. Li, and J. Mu, Mobile traffic prediction in consumer applications: A multimodal deep learning approach, *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3425–3435, 2024.

[40] B. Hussain, M. K. Afzal, S. Ahmad, and A. M. Mostafa, Intelligent traffic flow prediction using optimized GRU model, *IEEE Access*, vol. 9, pp. 100736–100746, 2021.

[41] S. Saha, A. Haque, and G. Sidebottom, Transfer learning based efficient traffic prediction with limited training data, in *Proc. 20th Consumer Communications & Networking Conference* (*CCNC*), Las Vegas, NV, USA, 2023, pp. 477–480.

[42] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, A comprehensive survey on transfer learning, *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[43] S. Zhang, T. Li, S. Hui, G. Li, Y. Liang, L. Yu, D. Jin, and Y. Li, Deep transfer learning for city-scale cellular traffic generation through urban knowledge graph, in *Proc. 29th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Long Beach, CA, USA, 2023, pp. 4842–4851.

[44] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, Transfer learning for time series classification, in *Proc. 2018 IEEE Int. Conf. Big Data* (*Big Data*), Seattle, WA, USA, 2018, pp. 1367–1376.

[45] 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 J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, 2019.

[46] A. Dridi, H. Afifi, H. Moungla, and C. Boucetta, Transfer learning for classification and prediction of time series for next generation networks, in *Proc. ICC 2021-IEEE Int. Conf. Communications*, Montreal, Canada, 2021, pp. 1–6.

[47] Q. Zeng, Q. Sun, G. Chen, H. Duan, C. Li, and G. Song, Traffic prediction of wireless cellular networks based on deep transfer learning and cross-domain data, *IEEE Access*, vol. 8, pp. 172387–172397, 2020.

[48] Q. Wu, K. He, X. Chen, S. Yu, and J. Zhang, Deep transfer learning across cities for mobile traffic prediction, *IEEE ACM Trans. Netw.*, vol. 30, no. 3, pp. 1255–1267, 2022.

[49] H. Li, J. Wang, X. Chen, X. Liu, and G. Dudek, Data-efficient communication traffic prediction with deep transfer learning, in *Proc. ICC 2022-IEEE Int. Conf. Communications*, Seoul, Republic of Korea, 2022, pp. 3190–3195.

[50] X. Wan, H. Liu, H. Xu, and X. Zhang, Network traffic prediction based on LSTM and transfer learning, *IEEE Access*, vol. 10, pp. 86181–86190, 2022.

[51] A. Petrella, M. Miozzo, and P. Dini, Mobile traffic prediction at the edge through distributed and deep transfer learning, *IEEE Access*, vol. 12, pp. 191288–191303, 2024.

[52] C. Parera, Q. Liao, I. Malanchini, D. Wellington, A. E. C. Redondi, and M. Cesana, Transfer learning for multi-step resource utilization prediction, in *Proc. 31st Ann. Int. Symp. Personal, Indoor and Mobile Radio Communications*, London, UK, 2020, pp. 1–6.

[53] X. Chen, G. Chuai, K. Zhang, and W. Gao, Spatial-temporal cellular traffic prediction: A novel method based on causality and graph attention network, in *Proc. 2023 IEEE Wireless Communications and Networking Conf.* (*WCNC*), Glasgow, UK, 2023, pp. 1–6.

[54] J. Wang, W. Fan, C. Hu, and X. Zhang, User traffic collection and prediction in cellular networks: Architecture, platform and case study, in *Proc. 4th IEEE Int. Conf. Network Infrastructure and Digital Content*, Beijing, China, 2014, pp. 414–419.

[55] X. Xu, S. Gao, and Z. Jiang, LSTCN: An attention-based deep neural network model combining LSTM and TCN for cellular network traffic prediction, in *Proc. 5th Int. Conf. Communication and Information Systems* (*ICCIS*), Chongqing, China, 2021, pp. 34–38.

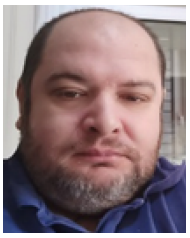
[56] A. Cao, Y. Qiao, K. Sun, H. Zhang, and J. Yang, Network traffic analysis and prediction of hotspot in cellular network, in *Proc. 2018 Int. Conf. Network Infrastructure and Digital Content* (*IC-NIDC*), Guiyang, China, 2018, pp. 452–456.

[57] M. Nakip, B. C. Gül, V. Rodoplu, and C. Güzeliş, Predictability of Internet of Things traffic at the medium access control layer against information-theoretic bounds, *IEEE Access*, vol. 10, pp. 55602–55615, 2022.

[58] N. Zhao, Z. Ye, Y. Pei, Y. C. Liang, and D. Niyato, Spatial-temporal attention-convolution network for citywide cellular traffic prediction, *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2532–2536, 2020.

[59] B. Yan, G. Wang, J. Yu, X. Jin, and H. Zhang, Spatial-temporal chebyshev graph neural network for traffic flow prediction in IoT-based ITS, *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9266–9279, 2022.

[60] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv: 1412.3555, 2014.



**Iacovos I. Ioannou** is a researcher at the Networks Research Laboratory, University of Cyprus, Nicosia, Cyprus. He is also a junior researcher at the Smart Networked Systems Research Group of the RISE Center. He received the associate degree in computer science from Cyprus College in 2001, the BSc in computer science from the University of Cyprus in 2006, the MS degree in computer and network security from Open University of Cyprus, Nicosia, Cyprus, in 2017, and the PhD degree from University of Cyprus, Nicosia, Cyprus, in 2021. His research interests include mobile and wireless communications, next-generation networks (5G) and device-to-device (D2D) communications, using artificial intelligence techniques. He is a highly skilled developer with 20 years of hands-on working experience in the information technology (IT) industry, ranging the spectrum of IT systems from analysis, development, installation, and management. He has worked at the Philelefthеros Publishing Group as IT administrator and programmer for seven years. He worked for the Cyprus stock exchange as IT and programmer for seven years and he worked at Primetel for six years as IT and software engineer at services department. He has vast experience with cellular network infrastructure and all modern development platforms and languages. He has a certificate in ICONIX/SCRUM and he is also CCNET certified from CISCO.

**Marios Lestas** received the BA and MEng degrees in electrical and information engineering from the University of Cambridge, Cambridge, UK, in 2000, and the PhD degree in electrical engineering from University of Southern California Los Angeles, CA, USA, in 2006. He is currently an associate professor with Frederick University, Nicosia, Cyprus. He has participated in a number of projects funded by the Research Promotion Foundation and the Europe. His research interests include the application of non-linear control theory and optimization methods in complex networks such as computer networks, transportation networks, power networks, molecular nanonetworks, and metasurfaces. In the aforementioned networks, he has investigated issues pertinent to congestion control, information dissemination, network vulnerability, demand response, and more recently privacy and security.

**Waqar A. Aziz** is pursuing the PhD degree at Department of Computer Science at the University of Cyprus, Nicosia, Cyprus. He received the BEng degree in electrical engineering from University of Engineering and Technology, Peshawar, Pakistan, in 2016, and the MS degree in electrical engineering from National University of Sciences and Technology, Islamabad, Pakistan, in 2020. His research interests encompass data-driven AI/ML-assisted wireless communication systems, metasurfaces, data centers, network optimization, and edge caching.

**Vasos Vassiliou** is currently an associate professor with Department of Computer Science, University of Cyprus, Nicosia, Cyprus, and the co-director of the Networks Research Laboratory (NetRL), University of Cyprus. Since November 2017, he has been the group leader at the Smart Networked Systems Research Group, RISE Center of Excellence on Interactive Media, Smart Systems and Emerging Technologies, Nicosia, Cyprus. He has also been appointed as the senate of University of Cyprus to the Board of Directors of CYNET, the National Research and Educational Network, where he serves as the chair, since November 2016. His research interests include protocol design and performance aspects of networks (fixed, mobile, and wireless), in particular mobility management, quality of service (QoS) adaptation and control, resource allocation techniques, wireless sensor networks, and the Internet of Things. He is a senior member of the ACM, the treasurer of the IEEE Cyprus Section, the chair of the IEEE ComSoc Cyprus Chapter, part of the IBM Academic Initiative, and the academic advocate for the ISACA Cyprus Chapter. He is also on the editorial boards of *Telecommunication Systems* and *Computer Networks*, among others. He has been a member of the TPC for about 100 international conferences in his field and acted as the chair of various committees in more than 15 conference organizations.