



HAL
open science

Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions

Ons Aouedi, Le van An, Kandaraj Piamrat, Ji Yusheng

► **To cite this version:**

Ons Aouedi, Le van An, Kandaraj Piamrat, Ji Yusheng. Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions. ACM Computing Surveys, 2024, pp.1-35. hal-04811081

HAL Id: hal-04811081

<https://hal.science/hal-04811081v1>

Submitted on 29 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions

ONS AOUEDI, SnT, SIGCOM, University of Luxembourg, Luxembourg

VAN AN LE, the National Institute of Advanced Industrial Science and Technology (AIST), Japan

KANDARAJ PIAMRAT, Nantes Université, CNRS, INRIA, LS2N, UMR 6004, F-44000, Nantes, France

YUSHENG JI, National Institute of Informatics, Tokyo, Japan

From the perspective of telecommunications, next-generation networks or beyond 5G will inevitably face the challenge of a growing number of users and devices. Such growth results in high-traffic generation with limited network resources. Thus, the analysis of the traffic and the precise forecast of user demands is essential for developing an intelligent network. In this line, Machine Learning (ML) and especially Deep Learning (DL) models can further benefit from the huge amount of network data. They can act in the background to analyze and predict traffic conditions more accurately than ever, and help to optimize the design and management of network services. Recently, a significant amount of research effort has been devoted to this area, greatly advancing network traffic prediction (NTP) abilities. In this paper, we bring together NTP and DL-based models and present recent advances in DL for NTP. We provide a detailed explanation of popular approaches and categorize the literature based on these approaches. Moreover, as a technical study, we conduct different data analyses and experiments with several DL-based models for traffic prediction. Finally, discussions regarding the challenges and future directions are provided.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computer science** → *Deep learning*; • **Networks traffic analysis** → Traffic prediction.

Additional Key Words and Phrases: Deep Learning, Machine Learning, Network Traffic Prediction, network management.

ACM Reference Format:

Ons Aouedi, Van An Le, Kandaraj Piamrat, and Yusheng Ji. 2018. Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions. 1, 1 (May 2018), 35 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

The huge number of smart devices made the Internet widely used and accordingly triggered a surge in traffic and applications. This explosion increases the complexity of the network and the amount of data that needs to be collected and managed [67]. On the way to fully automated network management, one of the essential problems lies in accurate traffic prediction. Network Traffic Prediction (NTP) aims to forecast the total amount of traffic expected based on historical data to avoid future congestion and maintain high network quality [64]. It enables the network operator to present resource-allocation strategies; and in turn, optimizes the network resources dynamically.

Authors' addresses: Ons Aouedi, ons.aouedi@uni.lu, SnT, SIGCOM, University of Luxembourg, Luxembourg; Van An Le, the National Institute of Advanced Industrial Science and Technology (AIST), Japan, an.le@aist.go.jp; Kandaraj Piamrat, Nantes Université, CNRS, INRIA, LS2N, UMR 6004, F-44000, Nantes, France, kandaraj.piamrat@ls2n.fr; Yusheng Ji, National Institute of Informatics, Tokyo, Japan, kei@nii.ac.jp.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

XXXX-XXXX/2018/5-ART \$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

NTP can be treated as a time series forecasting problem and according to the solving methods, existing works can be roughly divided into two categories: (i) *statistical-based* methods and (ii) *Machine Learning (ML) or Deep Learning (DL)* models. Generally, most of the statistical-based methods are linear statistical methods (e.g., Auto-Regressive Integrated Moving Average - ARIMA) and are applied to lots of practical scenarios. However, such simple methods are inadequate for traffic prediction, taking into account the complex traffic patterns in realistic scenarios [36]. In particular, these methods can perform well for short-term forecasts but their performance deteriorates severely for long-term predictions [88]. Moreover, they require stable data and a small dataset [56] [116]. Consequently, it becomes increasingly clear that linear models are not adapted to complex applications [27]. On the other hand, with the surge of the network traffic data and the advances in ML and DL models [103], data-driven ML-based traffic prediction methods have established themselves as strong competitors to classical statistical models and obtained tremendous attention in the network domain. However, shallow learning algorithms such as linear regression or support vector regression may not perform well due to their limited parameter space in modeling complex network traffic. In other words, the spatial and/or temporal dependencies of the traffic should be modeled to improve performance [122]. Thus, as the network traffic is highly nonlinear from both temporal and spatial dimensions, it predicts future traffic volume a very challenging task and beyond the ability of linear models. In this context, DL can be suitable to infer information from large datasets and requires very little domain knowledge and engineering by hand [52]. DL has achieved great success in many applications including NTP [130] in comparison to shallow ML models because computers today have the computational power to build complex models that can process and learn from big data. Consequently, DL enables network topology to self-optimize, improve efficiency, and in turn, lead to more stable network connections for end-users and businesses.

This paper focuses on highlighting the application of DL-based models and techniques for NTP, presenting an extensive review of the taxonomy of DL-based models and the state-of-the-art approaches proposed by using such models. In addition, data analysis and an empirical evaluation of the performance of representative DL models for NTP have been performed.

Table 1. Summary of related reviews on deep learning methods for network traffic prediction

Ref.	Contributions	DL-based technique	NTP	Practical evaluation
[17]	A comprehensive survey on the application of shallow ML models for network traffic classification, traffic prediction, and intrusion detection.	x	*	x
[111]	A survey on the ML/DL for traffic classification, resource management, QoS/QoE prediction, and routing optimizing in an SDN environment.	✓	x	x
[91]	An overview on only the ML techniques for detecting network intrusions in SDN.	✓	*	*
[29]	A brief review on the application of shallow ML model for NTMA.	x	*	x
[2]	A comprehensive survey on the application of DL for NTMA.	✓	*	x
[99]	A review on the application of unsupervised learning in the domain of networking such as intrusion detection and traffic classification.	✓	x	x
[121]	A survey on DL for mobile and wireless networking.	✓	*	x
[10]	A survey on the application of ML/DL in software environments for network traffic management.	✓	*	x
[63]	A review from statistical to machine learning-based NTP.	x	✓	x
[16]	A brief review on ML techniques in time series forecasting.	✓	x	x
[4]	A comparison study for the major ML-based models for time series forecasting.	x	x	✓
[25]	A review on the linear and DL methods dedicated to time series analysis.	✓	x	x
[107]	A review and taxonomy of data augmentation methods for time series.	x	x	x
[48]	A review on the studies on cellular traffic prediction.	*	*	✓
Our study	A comprehensive review of DL-based models and techniques for NTP as well as an empirical evaluation of the performance.	✓	✓	✓

✓, x, and * indicate that the topic is total, not, or partially covered respectively.

1.1 Related work and our key contributions

Recently, several research efforts have been made to survey the use of DL and shallow ML models for network traffic, e.g., traffic classification or intrusion detection systems. In this context, Boutaba *et al.* [17] surveyed shallow ML models for network traffic classification and NTP. Xie *et al.* [111] surveyed ML/DL models used for traffic classification, resource management, Quality of Service/Experience (QoS/QoE) prediction, and routing optimization in a Software Defined Networking (SDN) environment. Whereas Sultana *et al.* [91] focused only on ML models for detecting network intrusions in SDN. Alconzo *et al.* [29] focused on the big data approach for Network Traffic Monitoring and Analysis (NTMA). They briefly discussed big data analytics (e.g., shallow ML models) for NTMA applications (i.e., traffic classification, traffic prediction, fault management, and network security). In the same context, Abbasi *et al.* [2] proposes a comprehensive survey on the application of DL for NTMA. Similarly, Aouedi *et al.* [10] proposed a survey on the application of ML/DL in software-defined environments for network traffic analysis, including traffic classification, prediction, and anomaly detection. Moreover, Usama *et al.* [99] studied the application of unsupervised learning in the domain of networking such as intrusion detection and traffic classification. Furthermore, Pacheco *et al.* [70] proposed a review paper to summarize the used steps that help to achieve traffic classification using ML models. Another work proposed by Zhang *et al.* [121] reviewed the application of DL for the mobile network including NTP. Lohrasbinasab *et al.* [63] provided a review of statistical to machine learning-based network traffic prediction. Last but not least, Jiang *et al.* [48] proposed a review of statistical and ML/DL-based approaches only on cellular traffic prediction. The comparison of the related works and our paper is summarized in Table 1.

Although these papers review the application of DL for intelligent network traffic analysis and highlight challenges and future directives for the use of DL, they do not treat or partially cover the problem of NTP. For this reason, some of the recent works [48] [63] [32] proposed a survey on NTP approaches; however, they considered only part of the traffic prediction studies, either only statistical/shallow models for NTP or DL models for only cellular traffic prediction. More specifically, the potential of the taxonomy of DL-based models such as Transfer Learning (TL), Federated Learning (FL), and Graph Neural networks (GNN) has not been explored in the literature. In contrast to such works, we present state-of-the-art DL-based models and techniques for NTP including Internet, edge, and cellular traffic. In particular, different from existing works, this paper aims to illustrate how DL can be used to model time-series data collected and provide an accurate prediction in different network scenarios. Also, this paper provides a data analysis of the widely used datasets and it covers the available datasets and some open-source code/datasets to facilitate reproducibility (to support further research). Furthermore, the open research challenges in our paper are significantly different than those in [48] [63]. In addition, we provide an experimental analysis of the most popular DL-based models for NTP. To the best of our knowledge, this paper is one of the first papers that investigate such a problem. In brief, the contributions of this survey can be summarized as follows:

- We provide a comprehensive and recent review of DL-based models and techniques for NTP purposes.
- We provide analysis for some reference datasets in the temporal and spatial domains.
- We provide an experimental analysis of different DL-based models. For each DL model, we study the performance and complexity under three reference datasets, namely, Abilene, GÉANT, and SDN datasets.
- We highlight several important issues and challenges associated with current studies on DL for NTP.

1.2 Paper Organization

The remainder of the paper is organized as follows. Section 2 presents the main DL models and their strengths/weaknesses, and then we present the comparison and summarized table for the reviewed DL models. The main purpose of Section 3 is to provide a comprehensive overview of DL applications for the prediction task. Section 4 lists and analyzes commonly used datasets in NTP, as well as provides links to open-source codes in the investigated papers. Section 5 provides useful information on the performance of widely used DL models in NTP. Section 6 introduces open issues and future directions. Finally, Section 7 concludes the paper.

2 BACKGROUND

In this section, the NTP problem is presented, followed by a summarized table of representative DL-based models, as well as common evaluation metrics.

2.1 Network Traffic Prediction (NTP)

NTP is a critical task aimed at predicting future traffic flow to improve QoS, mitigate network congestion, and enhance several network management applications, such as anomaly detection and bandwidth location[65]. Unlike conventional time-series tasks, NTP faces unique challenges due to the highly dynamic and complex nature of network environments. Below, we outline some of the key problem-specific challenges that make NTP distinct:

- **Irregular and Non-stationary Traffic Fluctuations:** The increasing number of smart devices and heterogeneous traffic sources contribute to irregular traffic patterns, making prediction more complex. These fluctuations often cause large prediction errors, potentially violating Service Level Agreements (SLAs). For example, during the COVID-19 pandemic, global Internet traffic surged due to remote work, online learning, and increased use of streaming services. This sudden, unprecedented rise in traffic created irregular patterns that traditional models could not predict accurately. NTP models must account for such external events, which are not typically seen in other time-series prediction tasks. Similarly, natural catastrophes, such as hurricanes or earthquakes, can cause sudden traffic surges due to emergency communication needs, drastically changing traffic patterns in affected areas and requiring real-time adjustments in the network. These events introduce high variability that challenges the robustness of prediction models.
- **Spatiotemporal Dependencies:** Network traffic is influenced by both temporal and spatial correlations. Traffic at one point in the network can influence traffic elsewhere, making it essential for prediction models to capture these interdependencies. Unlike traditional time-series prediction tasks, which are primarily focused on temporal relationships, NTP must consider the complex topology of networks and the traffic flows across multiple interconnected nodes [131]. Predicting traffic at different scales also adds complexity, as traffic patterns can vary widely depending on the time or day, week, or season. For example, during COVID-19, work-from-home setups created new traffic spikes at residential locations during business hours, altering both temporal and spatial traffic patterns. Natural catastrophes can also alter network traffic by redirecting communication needs to specific areas or affecting the infrastructure, leading to sudden shifts in traffic load across different regions. These spatiotemporal dependencies require NTP models to be both adaptable and context-aware.
- **Impact of External Factors:** Network traffic is influenced by several external factors such as the number of base stations (BSs), geographical factors, and even weather conditions like temperature and humidity. These factors can further complicate predictions and require models to integrate multimodal data sources. The COVID-19 pandemic provides a perfect

example of how external events can disrupt normal traffic behavior. With restrictions on movement, there was a sudden drop in traffic from commercial areas and a surge in residential network use, which altered traditional traffic flow patterns. Unlike traditional time-series tasks, NTP must be robust enough to incorporate such external influences.

NTP can be formulated as predicting the future traffic volume ($\hat{y}_t + l$) based on historical and current traffic volumes ($X_{t-J+1}, X_{t-J+2}, \dots, X_t$). The model's objective is to find parameters that minimize the error between the predicted and observed traffic volumes, as shown in Equations 1 and 2. The challenge lies in selecting the right model parameters and accurately capturing the complex spatiotemporal patterns inherent in network traffic.

$$W^* = \operatorname{argmin} W^* L(y_{t+l}, \hat{y}_{t+l}; W^*) \quad (1)$$

$$\hat{y}_t + l = f([X_{t-J+1}, X_{t-J+2}, \dots, X_t]) \quad (2)$$

Here, y_{t+l} and \hat{y}_{t+l} are the observed and predicted values at time $t+l$, l is the prediction horizon, $f(\cdot)$ is the activation function, L is the loss function, and W^* is the optimal set of parameters.

There are two primary approaches for predicting traffic: link load prediction and network traffic matrix (TM) prediction [62].

- **Link Load Prediction:** This is treated as a univariate time-series problem, where the future traffic load of a specific link is predicted based solely on its historical values. While this method simplifies the problem, it assumes that each link is independent, which is an unrealistic assumption in real-world networks where traffic is interconnected.
- **Traffic Matrix (TM) Prediction:** A more comprehensive approach is predicting the TM, which considers the traffic volume across all links in the network. The TM captures the interdependencies between links, providing a more accurate representation of network traffic [97]. This approach acknowledges the interconnected nature of network traffic, making it more suitable for capturing the spatial dependencies in NTP.

Given the complexity of NTP, models must be capable of handling multimodal data inputs, including traffic volumes, spatial relationships, and external factors, while providing scalable solutions for large, distributed networks. Addressing these challenges requires models that go beyond traditional time-series prediction methods and integrate advanced techniques such as Graph Neural Networks (GNNs), attention mechanisms, and hybrid models to effectively capture the complex spatiotemporal dynamics of network traffic.

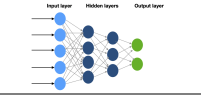
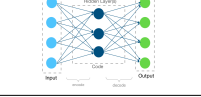
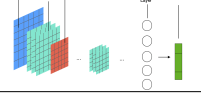
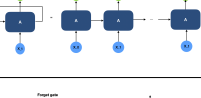
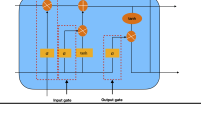
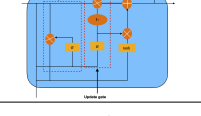
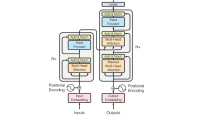
2.2 Taxonomy of DL models

DL, also known as Deep Neural Networks (DNN), represents one of the most active areas of Artificial Intelligence (AI) research [77]. It is a branch of ML that evolved from neural networks (NNs), enabling an algorithm to make predictions or classifications based on large datasets without being explicitly programmed. The major benefits of DL over shallow ML models are its superior performance for large datasets [126] and the integration of feature learning and model training in one architecture. In the literature, DL has also been referred to as deep structured learning, hierarchical learning, and deep feature learning [15]. Through its deep architecture, it has a higher learning capacity compared to shallow ML models and can accordingly learn highly complicated patterns [132]. It uses supervised and unsupervised learning to learn high-level features for the tasks of classification and pattern recognition.

From the recent literature, we can find representative DL models that are frequently used in prediction tasks, namely *Multilayer perceptron (MLP)*, *AutoEncoder (AE)*, *Convolutional neural network (CNN)*, *Recurrent neural network (RNN)*, *Long Short Term Memory (LSTM)*, and *Gate*

recurrent unit (GRU), and Transformers. Table 2 summarizes their description along with their strengths/weaknesses and generic structure.

Table 2. Summary of different deep-learning models used in NTP.

Models	Description	Strengths/Weaknesses	Structure of the model
MLP	MLP is a class of feedforward artificial neural networks (ANN), which consists of three or more layers. The first layer is for input data. One or more hidden layers extract features from the input. The last layer outputs a classification result.	<i>Strength:</i> Easy to implement; <i>Weakness:</i> Modest performance, slow convergence, occupies a large amount of memory.	
AutoEncoder	Autoencoders can be divided into three parts, which are encoder, code, and decoder blocks. The encoder obtains the input and converts it into an abstraction, and then the input can be reconstructed from the code layer through the decoder.	<i>Strength:</i> Works with large and unlabeled datasets, suitable for feature extraction and used instead of manually engineered extraction; <i>Weakness:</i> The quality of features depends on the model architecture and its hyper-parameters, and it is hard to find the code layer size.	
CNN	CNN is a class of DL, which consists of several convolutions and pooling (subsampling) layers followed by fully connected layers. It is widely used for image recognition applications.	<i>Strength:</i> Weights sharing, extracts relevant features, provides highly competitive performance; <i>Weakness:</i> High computational cost, requires large training dataset and a high number of hyper-parameters tuning to achieve optimal features.	
RNN	RNNs are neural networks that have one or more connections between neurons that form cycles. These cycles are responsible for storing and passing the feedback from one neuron to another.	<i>Strength:</i> Simple to implement, faster than LSTM and GRU, the ability to capture temporal behaviors; <i>Weakness:</i> When modeling long sequences, their ability to remember what they learned before many time steps may decline.	
LSTM	LSTM is an extension of RNNs. It has internal mechanisms called gates (forget gate, input gate, and output gate) that can learn which data in a sequence are important to keep or to throw away [42].	<i>Strength:</i> Good for sequential information, works well with long sequences; <i>Weakness:</i> High model complexity, high computational cost.	
GRU	GRU was proposed in 2014. It is similar to LSTM with fewer parameters. Unlike LSTM, GRU has two gates, which are the update gate and the reset gate, and therefore it is less complex [21].	<i>Strength:</i> Computationally more efficient than LSTM; <i>Weakness:</i> Less accurate than LSTM.	
Transformers	Transformers are characterized by their attention mechanism, which enables them to weigh the importance of different input elements when generating predictions.	<i>Strength:</i> Capture long-range dependencies, process sequences in parallel; <i>Weakness:</i> Require significant computational resources and memory.	

2.3 Criteria of prediction performance

For a better understanding of prediction performance, the most frequently used evaluation metrics are provided: *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)*, *Root Mean Square Error (RMSE)*, *Normalized Root Mean Squared Error (NRMSE)*, and *Mean Average Percentage Error (MAPE)*. These metrics are used to measure the prediction error and hence model performance: the smaller the result, the better the model. R^2 measures the ability of the predicted result to represent the actual data: the larger the value, the better the prediction effect.

- *Mean Squared Error (MSE)*: calculates the average of the squared differences between predicted and actual values. It penalizes large errors more significantly than MAE, making it sensitive to outliers.

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (3)$$

- *Root Mean Squared Error (RMSE)*: is the square root of MSE and provides a measure of the standard deviation of prediction errors. It is expressed in the same units as the predicted

values, allowing for easier interpretation.

$$RMSE = \sqrt{MSE} \tag{4}$$

- Mean Absolute Error (*MAE*): measures the average magnitude of errors between predicted and actual values. It provides a straightforward indication of the average prediction error without considering the direction of errors.

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \tag{5}$$

- Mean Average Percentage Error (*MAPE*): calculates the average percentage difference between predicted and actual values. It offers insights into the relative accuracy of predictions and is particularly useful for interpreting errors in terms of their percentage of the actual values.

$$MAPE = \frac{1}{N} \sum_{i=1}^N 100 \times \frac{|\hat{y}_i - y_i|}{|y_i|} \tag{6}$$

- Coefficient of Determination (R^2): measures the proportion of the variance in the dependent variable (actual values) that is predictable from the independent variable (predicted values). It ranges from 0 to 1, where 1 indicates perfect prediction. R^2 provides insights into how well the model fits the observed data and is particularly useful for assessing the overall goodness of fit of the model.

$$R^2 = 1 - \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{\sum_{i=1}^N (\hat{y}_i - \bar{y})^2} \tag{7}$$

- Normalized Root Mean Squared Error (*NRMSE*): provides a relative measure of predictive accuracy that accounts for the variability in the data. *NRMSE* allows for comparison across different datasets with varying scales, making it useful for assessing model performance in a more generalized manner.

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{8}$$

3 DL APPLICATIONS IN NETWORK TRAFFIC PREDICTION

This section presents relevant works on NTP that use DL-based models. As shown in Fig. 1, we have divided the existing work into six categories: *Simple DNNs*, *Hybrid DNNs*, *Multi-task Learning*, *Federated Learning*, *Transfer Learning*, and *Graph Neural Network*. We summarize the papers investigated with respect to the contribution, model category, network level, spatial/temporal data, and the dataset used.

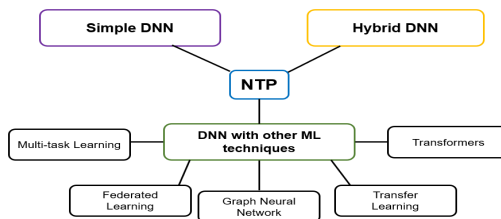


Fig. 1. Deep Learning for Network Traffic Prediction.

3.1 Simple Deep Neural Networks

3.1.1 *Long Short Term Memory (LSTM)*. LSTM is a famous time series prediction model based on DL. It was introduced to overcome the problem of the RNN model. With the help of information gates, LSTM is capable of capturing the long-range dependency of network traffic.

In this context, Azari *et al.* [11] presented a comparative study of LSTM and ARIMA. They analyzed the impact of different parameters on the effectiveness of the predictions. The results demonstrate the superior performance of LSTM over ARIMA models in cellular traffic, especially when the training time series is longer. Similarly, Jaffry *et al.* [47] designed an LSTM-based cellular traffic prediction model. The comparative analysis of the ARIMA and Feed Forward Neural Networks (FFNN) models, demonstrates that LSTM learns the traffic patterns very quickly even with a small amount of training data sample, but it needs more time for the training task. In addition, Trinh *et al.* [94] compared LSTM to MLP and ARIMA and showed that LSTM significantly outperforms the rest of the models. Azzouni *et al.* [12], also proposed a framework called *neuTM* to learn the characteristics of the traffic from historical traffic data and predict the future TM using LSTM. The LSTM model was deployed on software-defined networking (SDN) and trained on a real-world dataset using different configurations. The experimental results showed that *neuTM* can outperform linear forecasting models and FFNN models. Feng *et al.* [31] proposed a new Deep Traffic Predictor, called (*DeepTP*), which forecasts traffic demand from spatially dependent and temporal cellular traffic. It consists of two components: (i) a general feature extractor for modeling spatial dependencies and encoding the external information and (ii) a sequential module for modeling the temporal variations. For the general feature extractor, they introduced a correlation selection mechanism for spatial modeling and an embedding mechanism to encode external information (POI -Point of Interest category and the day of the week). Then, because the LSTM is not accurate enough due to the brittleness and complexity, they applied the *Seq2Seq* model [92] with the *attention mechanism* [13] to build the sequential model. The results show that the introduction of external information can increase the performance of traffic prediction.

Additionally, to improve resource management in the 5G network and improve transmission bandwidth and communication latency, Alawe *et al.* [5] proposed an LSTM-based mechanism to anticipate traffic load changes in the 5G network. Specifically, traffic prediction values enable dynamic scaling of 5G network resources, particularly Access and Mobility Management (AMF) resources. The simulation results showed that the forecast-based scalability mechanism outperforms threshold-based solutions in terms of latency (to react to traffic change) and delay to have new resources ready to be used by the Virtual Network Function (VNF) to react to traffic increase.

Although LSTM performs well in comparison to the RNN model, it requires a high computational cost for training, and its computing time is proportional to the number of parameters. To handle these issues, Hua *et al.* [45] proposed a sparse LSTM model, called *RCLSTM*. The basic idea behind *RCLSTM* is to build an LSTM model with sparse neural connections. Using two datasets, *RCLSTM* performs a competitive prediction performance compared to LSTM and reduces computing time by 30%. In the same direction and to reduce the density of resources in LSTM, Xiong *et al.* [112] also proposed a sparse connected LSTM with a shared weight, called *SCLSTM*. In other words, they pruned the LSTM parameters, and then the sharing weight operation is used on the sparse weight matrix generated after the pruning strategy. The experimental results on the real dataset demonstrate that *SCLSTM* can reduce the weight storage of densely connected LSTM by 113.63 times.

3.1.2 *Convolution Neural Networks (CNN)*. CNN models are usually deployed to extract spatial features by decomposing the traffic network into grids and using the convolution operation [115] [123]. In this context, Zhang *et al.* [123] proposed a new method for predicting the traffic of the city's

network. They treated network traffic as an image by exploiting densely connected CNN to capture and predict the spatial and temporal dependencies of traffic. In addition, its framework fuses different types of temporal dependencies (i.e., closeness and period) using a parametric matrix-based fusion strategy. In other words, a convolution layer is added separately to the L(th) layer of the network to fuse the features of closeness and period. The experiments are carried out with two types of datasets (i.e., SMS and calls). Using *RMSE* as an evaluation metric, the results show that the proposed method achieves the most accurate prediction.

Furthermore, to improve the performance of the CNN-based model for NTP, Shen *et al.* [87] proposed a CNN scheme aided by time-of-attention, called *TWACNet*. More specifically, the time-wise attention mechanism, based on the self-attention mechanism, is used to capture long-range temporal features, whereas the CNN model is adopted to capture spatial features. Furthermore, *TWACNet* uses external characteristics, such as the number of BS, POI and social activities, to improve prediction performance. Using the Telecom Italia dataset, the experiment result demonstrated the effectiveness of the *TWACNet* scheme in terms of *RMSE* and training time. Table 3 summarizes the used models, datasets, the extracted features as well as the key contributions.

Table 3. Summary of Applications of Simple DL-based Models in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[12]	Cellular network	LSTM	Temporal	Demonstrate that LSTM performs better than the linear forecasting models and the feedforward neural network (FFNN).
[123]	Cellular network	Densely connected CNN	Both	Treats the traffic as images and use CNNs to model closeness and period temporal dependency.
[94]	Cellular traffic	LSTM	Temporal	Exploit the ability of the LSTM to enhance the prediction accuracy
[31]	Cellular network	Seq2Seq LSTM, Attention mechanism	Both	Extracts the spatial and temporal features using several modules.
[47]	Cellular network	LSTM	Temporal	Demonstrates that LSTM learns traffic patterns very quickly even with a small amount of training data sample.
[5]	5G network	LSTM	Temporal	Shows that LSTM performs more than 10% better than DNN and the forecast-based scalability mechanism outperforms the threshold-based solutions.
[11]	Cellular network	LSTM	Temporal	compare the performance of LSTM over ARIMA.
[45]	-	Sparse LSTM	Temporal	build an LSTM model with sparse neural connections.
[87]	Cellular network	CNN, Attention mechanism	Both	combines the Attention mechanism and the CNN model to capture spatial dependencies.
[112]	-	LSTM	Temporal	Reduce the complexity of LSTMs through pruning strategies.

3.2 Hybrid Deep Neural Networks

3.2.1 CNN-LSTM. The standard LSTM is efficient in handling temporal correlations but fails to extract spatial features with network traffic. To consider spatial correlations, some works have started to use the CNN model [66], which can find spatial correlations on the network map, made possible by using convolution layers. To overcome the limitations of LSTM and extract spatiotemporal features, CNN-LSTMs are proposed as a combination of CNN and LSTM layers. For example, Gao *et al.* [35] proposed a DL-based model for the prediction of TM by considering both interflow correlation (spatial characteristics) and intraflow dependencies (temporal characteristics), called *CRNN*. *CRNN* used the CNN model for the interflow correlation because the authors assumed that the traffic flows with the highest correlation share the same source or destination and, thus, are neighboring in TM. Then, for intra-flow correlation, they used LSTM to capture the temporal variations of traffic flows. Extensive experiments based on two real datasets showed that *CRNN* can significantly improve the prediction accuracy for future TM compared to state-of-the-art methods (ARIMA, SVR, LSTM and CNN).

Similar to [35], Gao *et al.* [34] focused on the prediction of TM considering spatial and temporal features. The only difference in this work is the use of an attention mechanism. Specifically, they proposed a novel Attention-based Convolutional Recurrent Neural Network, called *ACRNN*. The experimental results showed that compared to the state-of-the-art methods *ACRNN* can reduce *MSE* and *MAE* by 44.8% and 30.6%, respectively.

442 Furthermore, Jiang *et al.* [49] proposed a ConvLSTM model for Internet traffic matrix prediction,
443 called *ConvLSTM-TM*. Using three datasets (Abilene, CERNET, and GÉANT), *ConvLSTM-TM* out-
444 performs several DL-based models with a lower prediction error. In particular, using the proposed
445 model decreases the *RMSE* by 10.5%, 36.4%, and 56.1%, and decreases the *MAE* by 26.2%, 34.9%, and
446 61.3%.

447 In the same direction, Li *et al.* [56] proposed a DL-based framework by combining the residual
448 network, LSTM, and the attention mechanism, called *LA-ResNet*. *LA-ResNet* consists of 5 parts: 1) the
449 input of the model (responsible for inputting the processed traffic data), 2) ResNet (responsible for
450 extracting spatial features from the data traffic sequence), 3) the LSTM (responsible for extracting
451 temporal features from the data traffic sequence), 4) the attention module (based on the intermediate
452 output to improve the accuracy and stability of the prediction), 5) fully connected layer (responsible
453 for the output of the model prediction results). Using *RMSE* and average accuracy *MA* evaluation
454 metrics, the experimental results show that *LA-ResNet* outperforms some state-of-the-art approaches
455 such as ARIMA, 3DCNN, LSTM, GRU, CNN + RGN, and multitask learning (MTL [46]). Recently,
456 Wang *et al.* [106] combined LSTM, CNN-based models, and the attention mechanism to learn
457 the local short-term and long-term spatial-temporal features, called *RACnv*. *RACnv* consists
458 of ResConv3D and AConvLSTM modules and on a dataset collected from a Canadian wireless
459 service provider, Rogers Communications Inc. The results of the experiment show that the *RACnv*
460 network outperforms the ConvLSTM network with a different number of observations.

461 Since modern network communication is complicated, continually collecting all traffic from
462 the network is impractical. Thus, proposing the NTP approach under the constraint of incom-
463 plete/missing data is a promising solution. In such a context, Le Nguyen *et al.* [55] focused on the
464 prediction of future traffic in the backbone network under partial historical traffic data. The authors
465 applied ConvLSTM (a combination of CNN and LSTM) to extract the spatiotemporal features of the
466 TM data. Also, they used bidirectional ConvLSTM to update incorrect data in the input matrix to
467 improve prediction accuracy. In addition, they proposed a formula to determine which flows should
468 be measured in the future to reduce the monitoring overhead. The proposed approach performs
469 better than ARIMA and standard LSTM in terms of *RMSE*, *ER*, and R^2 .

470
471 3.2.2 *AE-LSTM*. Due to the feature extraction and dimensionality reduction capabilities of AE,
472 the researchers tried to combine it with other models such as LSTM. In fact, AE can find a better
473 representation of the data than the initial raw data (i.e., input data) itself, which can boost the
474 performance of the LSTM model. For example, Wang *et al.* [102] proposed a hybrid DL model for
475 the prediction of traffic load. This hybrid DL model consists of denoising SAE for spatial modeling
476 and LSTM for temporal modeling. The AE consists of one Global Stacked AutoEncoder (GSAE)
477 and multiple Local Stacked AutoEncoder (LSAE). When predicting cell traffic, historical data is
478 collected from both the cell itself and its neighboring cells. Each cell has its LSAE for representation
479 encoding. Meanwhile, a GSAE takes all the cell data and produces a global representation. The
480 local representation is concatenated with the global representation to produce spatial modeling
481 and will be passed on to the LSTMs for prediction. LSAE offers good representation of input data,
482 reduces data size, and supports parallelism (LSAE models are independent of each other) and
483 application-aware training (can train only LSAE models corresponding to cells of interest). The
484 dataset used in this work consists of data collected from a large LTE network China Mobile at Suzhu.
485 The experiment results demonstrate superior performance over those of the SVM and ARIMA
486 models.

487 Furthermore, Zeb *et al.* [118] proposed an AE model based on LSTM to predict network data
488 traffic on edge devices in a 6G network. The authors orchestrated the KubeFlow deployment using
489 the K8s master at the orchestration center to train the model on the collected time-series data.

490

Using $RMSE$ and R^2 as evaluation metrics, the results of the experiments show that the model can accurately predict the traffic.

Although the efficiency of the above-proposed approaches, the similarity of different types of cellular services (e.g., call, Internet) and regions were not considered, at the same time, making knowledge transfer inter-domains possible leads to a more precise and personalized model, and one way to achieve this is via Transfer Learning. Table 4 summarizes the used models, datasets, the extracted features as well as the key contributions.

Table 4. Summary of Applications of Hybrid DL-based Models in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[55]	Backbone network	Convolutional LSTM	Both	Predicts the traffic when the historical data are missing/uncompleted.
[106]	Cellular Network	CNN, LSTM, Attention mechanism	Both	Combines LSTM, CNN-based models, and attention mechanism to learn the local short- and long-term spatial-temporal features.
[35]	Cellular networks	CNN, LSTM	Both	Captures the interflow correlations and the intraflow dependencies in TMs.
[49]	Internet network	ConvLSTM	Both	Model the Internet traffic matrix prediction problem as a video prediction task.
[56]	Cellular network	ResNet, LSTM, Attention mechanism	Both	Extracts the relevant time and spatial features from the traffic.
[102]	Cellular network	SAE, LSTM	Both	Uses SAE model for spatial features and the LSTM model for temporal features.
[118]	6G network	AE, LSTM	Temporal	Uses the model to predict the data traffic inflow in the edge devices.

3.3 DNN with other ML techniques

3.3.1 Transformers. In recent years, transformers have emerged as a revolutionary architecture in the field of DL, revolutionizing various sequence modeling tasks [39]. Originally introduced for natural language processing, transformers have since been adapted and applied to a wide range of domains, including time series forecasting and NTP. Their ability to capture long-range dependencies and temporal patterns in sequential data makes them particularly well-suited for modeling complex relationships in network traffic datasets. Unlike traditional RNNs and CNNs, transformers operate in a parallelizable manner, allowing efficient training and inference on large-scale datasets.

For instance, Chen *et al.* [20] combined multitask learning with the Transformer architecture and proposed the MTL-Trans model for time series modeling and multidimensional time series prediction. The results of the experiment show that MTL-Trans significantly improved the state-of-the-art results in the prediction of multitask time series. Liu *et al.* [61] proposed a spatial transformer block (STB) and a temporal transformer block (TTB) and then proposed an end-to-end spatial-temporal transformer ST-Tran combining STB and TTB. The ST-Tran can accurately extract temporal and spatial features simultaneously in short time intervals to achieve an effective prediction of cellular traffic. Experimental results in the real-world public traffic data set show its effectiveness and demonstrate the usability of the transformer structure in the prediction of cellular traffic. However, one downside of the vanilla transformer is that the self-attention mechanism requires the computation of a similarity measure for all pairs of regions throughout an entire city. This requires high computational complexity for a large city, which will increase non-linearly as the number of regions increases. Therefore, to reduce the computational complexity in the spatial domain and achieve better performance in all regions, Hu *et al.* [44] proposed a novel framework, STD-Net. STD-Net can extract complicated spatial, local, and global computations using a spatio-temporal-temporal transformer. It also uses a downsampling transformer to extract global spatial features from the entire city. Using real-world mobile traffic data sets (Telecom Italia), the proposed shows a superior prediction performance, while computational complexity analysis ensured that the cost of STD-Net remained quadratic, as in the transformer. Unlike previous studies, Zhang *et al.*

al. [128] integrated convolutional and self-attention mechanisms within a unique hybrid encoder and used a two-stage decoder to handle the high dynamic range intrinsic to spatio-temporal data to capture adjacent space-time relationships. Table 5 summarizes the used models, datasets, the extracted features as well as the key contributions.

Table 5. Summary of Applications of Transformers in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[20]	Cellular network	Transformers	Temporal	Combines transformers and multitask learning for multidimensional NTP.
[61]	Cellular network	Transformers	Both	Proposes end-to-end spatial-temporal transformer by combining spatial transformer and spatial transformer blocks.
[44]	Cellular network	MLP, CNN, Transformers	Both	Extracts complicated spatial, local, and global computations using a spatio-temporal-temporal transformer.
[128]	Cellular network	CNN, Transformers	Both	Integrates convolutional and self-attention mechanisms within a unique hybrid encoder and used a two-stage decoder.

3.3.2 *Deep Transfer Learning (DTL)*. TL tries to transfer the knowledge from the source domain to the target domain, as illustrated in Fig. 2. It can help the prediction model avoid learning from scratch, thus accelerating the convergence of the model and solving the problem of insufficient training data [93].

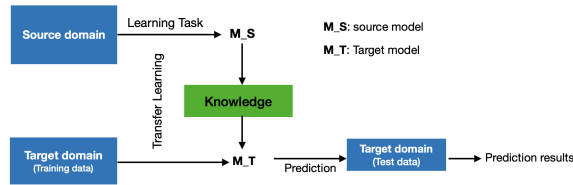


Fig. 2. Knowledge transfer between source and target domain

For example, Patil *et al.* [71] used the TL technique with the GRU model to handle the problem of insufficient IoT traffic data. The results demonstrate that the proposed *GRU-NN* model outperforms the other traffic predictors. In the same direction, Zeng *et al.* [119] focused on the impact of large cross-domain datasets and TL on traffic prediction accuracy. Taking the spatiotemporal cross-domain neural network (*STC-N*) as the benchmark model, different types of large datasets and TLs cross-domain were taken as research objects to discuss their influences on prediction performance. For TL, the K-means clustering method is used to group similar regions and transfer knowledge among them. Also, their model not only uses the similarity of different regions but also can transfer knowledge between different services (e.g., calls and SMS). The authors demonstrated that the more types of cross-domain datasets considered, the better the training performance of the model. Moreover, the TL technique increases prediction performance and gives better results than the model without an TL.

Furthermore, Zhang *et al.* [122] proposed a novel DL architecture termed the spatial-temporal cross-domain neural network in cellular networks (extension of [123]), called *STCNet*. *STCNet* used a convolutional LSTM network as a subcomponent to capture spatial-temporal dependencies. They predict traffic by capturing the complex patterns hidden in the data, metadata, and external factors that affect traffic generation. The cross-domain datasets (ie BS information, POI distribution, and social activity) were collected and modeled through *STCNet* to capture external factors. Then, a clustering algorithm partitioned the city areas into groups, and interclustering TL is introduced to improve the prediction performance. The wireless traffic data used in this work is coming from

589 Telecom Italia (a publicly available dataset). Finally, they used linear regression (LR), SVR, LSTM,
590 and DenseNet as baselines.

591 Moreover, Dridi *et al.* [28] proposed a TL-based model for cellular networks for predicting time
592 series. The proposed model was used for two cases, which are intra- and inter-cell. The intracell is
593 when the source and the target domain belong to the same cell, and the intercell is the use of TL
594 between two different cells. Using a real dataset, the results demonstrate the ability of TL to solve
595 the problem of insufficient amounts in the target domain. Also, it shows that using TL with intracell
596 performs better than on intercell. Recently, the *CCTP* framework was proposed for the prediction
597 of city-wide mobile traffic [108]. First, they presented a novel spatial-temporal learning model and
598 pre-train it with the source city (Milano) data to obtain prior knowledge of mobile traffic dynamics.
599 Then they applied a GAN-based approach to solving the domain shift problem due to a different
600 data distribution between the source and target domains. Finally, to deal with the data scarcity
601 issues in some clusters of the target city (Trentino), they further designed an intercluster TL strategy
602 for performance enhancement. The results show that TL can reduce prediction error and help
603 the model converge much faster than a model that is created from scratch. In the same direction,
604 Saha *et al.* [86] evaluated the performance of TL in the prediction of real world Internet traffic for
605 the network with smaller training data. The results show that TL improves model accuracy and
606 target domain learning became faster with the TL approach than standard learning for most target
607 domain datasets. Table 6 summarizes the used models, datasets, the extracted features as well as
608 the key contributions.
609

610 Table 6. Summary of Applications of Deep Transfer Learning-based Techniques in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[71]	IoT traffic	GRU	Temporal	Combine the GRU model and the transfer learning technique to handle the problem of insufficient IoT traffic data.
[122]	Cellular network	densely connected CNN, LSTM, K-means	Both	Uses Transfer Learning to predict traffic by capturing complex patterns hidden in data, metadata, and the external factors that affect traffic generation.
[28]	Cellular network	CNN, LSTM	Both	Evaluates the performance of Transfer Learning with intra- and intercell.
[119]	Cellular network	CNN, LSTM, FFNN	Both	Focuses on the impact of big datasets from the cross-domain and transfer learning on traffic prediction.
[108]	Cellular network	LSTM, GNN, GAN, Attention mechanism	Both	Presents a cross-city deep TL framework named <i>CCTP</i> for mobile traffic prediction in data-scarce city.
[86]	Internet network	LSTM, Autoencoder, Attention mechanism	Temporal	Evaluates the performance of TL in real-world internet traffic prediction for the network with smaller training data.

620
621 **3.3.3 Multi-task Learning (MTL).** MTL aims to perform several learning tasks simultaneously
622 under the assumption that the tasks are not completely independent and that one can improve the
623 generalization learning of another. MTL is similar to TL. In other words, with MTL, the objective
624 is to improve the performance of all tasks and there is no difference between the tasks, whereas
625 the objective behind TL is to improve the performance of a target domain/task with the help of
626 the source task [127]. It is more efficient than several single-task models because an MTL dataset
627 contains data on all tasks and, therefore, can help the MTL model perform better [82].

628 In this context, Rago *et al.* [80] proposed an MTL approach that integrates both traffic classifica-
629 tion and prediction tasks. This approach consists of three main components (AE model, classifier,
630 and predictor) and is divided into two steps. The first step consists of AE training (Undercom-
631 plete/Seq2Seq architecture) for feature extraction and, in turn, allows the joint execution of the
632 classification and prediction tasks. The second step consists of training both the classifier and the
633 predictor using the set of characteristics extracted in the first step. The classification and prediction
634 are executed through the softmax layer and fully connected layer respectively. Using the MTL, it
635 becomes possible to classify the type of application and predict the radio utilization pattern during
636 a time interval. The comparison of the conventional single task model (which does not use the AE
637

model and tackles the classification and prediction tasks separately) demonstrates the effectiveness of the proposed MTL approach.

Also, Nie *et al.* [68] proposed an MTL mechanism in an Industrial Internet of Things (IIoT) environment. This mechanism predicts jointly the future TM and the link load traffic to improve the prediction performance. Using two open datasets (Abilene and GÉANT), the results demonstrate that using the link load as an additional task can increase the generalization performance of the model. Table 7 summarizes the used models, datasets, the extracted features as well as the key contributions.

Table 7. Summary of Applications of Multi-task Learning in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[80]	Edge network	AE, FCN	Temporal	Joint traffic classification and prediction.
[68]	Industrial IoT network	LSTM	Temporal	Predicts jointly the future TM and the load traffic of the link.

3.3.4 Graph Neural Network (GNN). A graph is represented as $G = (V, E)$, where V is the set of nodes and E is the set of edges. Let $v_i \in V$ denote a node and $e_{ij} = (v_i, v_j) \in E$ denote an edge pointing from v_j to v_i . The neighborhood of a node v is defined as $N(v) = \{u \in V | (v, u) \in E\}$. For further information on graph kernel methods, we refer the reader to [109]. GNNs explore the relationships between nodes (e.g., base stations) and leverage them to obtain more intelligent predictions [50].

The diffusion convolutional recurrent neural network (DCRNN) [58] and the Wavenet graph models (GWN) [110] are proposed as an improved version of the convolution operation to capture spatial correlations in non-Euclidean data and show a more efficient representation of the traffic structure. GWN combines graph convolution with dilated casual convolution to capture spatial-temporal dependencies. With GWN, each graph convolution layer addresses the spatial dependencies of the node information extracted by dilated casual convolution layers at different granular levels. DCRNN is one of the well-known DL-based models for traffic prediction problems, which uses a diffusion process during the training stage to learn the representations of spatial dependency. With DCRNN, the encoder and decoder contain two recurrent layers. For example, Andreoletti *et al.* [8] proposed a graph-based DL approach through the recurrent neural network convolutional, called *DCRNN* (originally developed to forecast road traffic [59]). *DCRNN* forecasts the traffic load on the links of a real backbone network and detects congestion events due to its ability to learn the topological information of the network. Their approach is based on the idea that the relation between two nodes can be represented as a diffusion process. The diffusion process gives important clues about the influence that each node exerts on all the others. To do this, they considered a telecom network composed of a set of nodes and a set of links. They used LSTM, CNN, CNN-LSTM and a fully connected neural network as baselines using *MAPE*, *MAE*, and *RMSE*, as well as evaluation metrics related to convergence speed. The results showed that *DCRNN* works better but it is the slowest approach compared to the baselines.

In the same direction, Wang *et al.* [104] proposed a solution that models the spatiotemporal dependency from inter-tower and in-tower traffic and predicts cellular traffic at a large scale, through a GNN. Their model outperforms the state-of-the-art approaches by 13.2% and 17.5% in terms of *MAE* and *MAPE*, respectively. Moreover, it outperforms the state-of-the-art approaches using different levels of traffic volumes. Beyond the accurate prediction achieved in their study, this work also demonstrates the potential of inference of specific social events to improve NTP.

Furthermore, since the traffic in a wireless network is influenced not only by the historical traffic and the dataset between domains, but also by the handover traffic from the base station, Zhao *et al.* [129] proposed the *STGCN-HO* model. *STGCN-HO* is a cellular traffic prediction model

687 that uses the handover graph. It uses the transition probability matrix of the handover graph to
688 improve the NTP. Also, the authors fuse features from auxiliary data (i.e., day of the week, the hour
689 in the day), spatial, and temporal domains by constructing the graph convolution and the gated
690 linear unit as one spatiotemporal convolution block (ST-block). They perform batch normalization,
691 dropout layers within each ST-block, and residual connections are added for neighbor ST-blocks to
692 resolve the vanishing gradient problem as well as to avoid overfitting and accelerate training. The
693 evaluation was done on real-world 4G LTE traffic data contributed by a major telecom company.
694 Furthermore, they used HA, ARIMA, LSTM and multitask LSTM as a baseline using *MAE*, *RMSE*,
695 *RRMSE*, and training time as evaluation metrics. The results show that *STGCN-HO* is better than
696 the baseline models at the cell level and base station level.

697 Although the above approaches show an accurate traffic prediction, they are based on future
698 traffic load for a city, an urban area, or a base station, which is vague for a fine-granular user-
699 level traffic prediction. To address this problem, a fine-grained prediction was proposed by Yu *et*
700 *al.* [117]. The authors presented a spatial-temporal fine granular user traffic prediction mechanism
701 for cellular networks, called *STEP*. Specifically, *STEP* is based on the integration of the graph
702 convolution network (GCN) [33] and the GRU model, to capture the Spatio-temporal features of the
703 individual user traffic. To evaluate the performance of the proposed solution, the authors collected
704 data from 10 volunteers over a month using a specialized data collection application. This dataset
705 consists of traffic statistics and volunteer geolocation data. The experiment results show that *STEP*
706 performs better than ARIMA, LSTM and GNN in terms of *RMSE*. Unlike the GCN that uses the
707 same weight for neighboring nodes, Wang *et al.* [105] introduced the attention mechanism to
708 set the appropriate weights for each node. In particular, the authors proposed a graph attention
709 network based on time series similarity for cellular traffic prediction, called *TSGAN*. They used
710 Dynamic Time Warping (DTW) to calculate the time-series similarity between the network traffic
711 of every two cells and Graph Attention Networks to extract the spatial features. Then, comparison
712 experiments were conducted on Telecom Italia and Abilene datasets over GNN and GRU models to
713 demonstrate the performance of *TSGAN*. Recently, with rapid development, satellite communication
714 has become one of the most important communication means today. In this context, Yang *et al.* [114]
715 proposed an NTP approach for satellite networks, called *GCN-GRU*. In particular, they used the
716 GCN model for spatial feature extraction whereas the temporal features are captured by GRU. The
717 simulation results demonstrate that using GCN for spatial feature extraction can boost prediction
718 performance as well as outperform a single GRU model. Table 8 summarizes the used models,
719 datasets, the extracted features as well as the key contributions.

720 Although the performance of the above models, they could not be an efficient solution with
721 sensitive data. This is because collecting the data in a central entity for model training is a crucial
722 step. Thus, to solve this issue, Federated Learning (FL) appeared as a valuable approach. With FL,
723 each agent of the system collaboratively trains a global model over the decentralized network.

724
725 **3.3.5 Federated Learning (FL).** Within the FL concept, the data is maintained where it was generated,
726 and no raw data gets exchanged. In other words, FL is a distributed ML concept in which data entities
727 E_i collaborate to jointly learn a global model (e.g., traffic prediction model) without sacrificing
728 the privacy of end users [95]. Fig. 3 illustrates N data entities $\{E_1, \dots, E_N\}$ and their respective data
729 $\{D_1, \dots, D_N\}$. Therefore, it is more scalable than the centralized DL training process and can be a
730 promising solution for future network generation. For further information on FL, we refer the
731 reader to [74] [60] .

732 In such a context, Zhang *et al.* [120] proposed a novel wireless traffic prediction framework called
733 *FedDA*, which is trained collaboratively by multiple BSs. First, they used a clustering strategy to
734 group all BSs (i.e. clients in the context of FL) into several clusters depending on considering both

735

Table 8. Summary of Applications of GNN-based Techniques in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[104]	Cellular network	GNN	Both	Models the Spatio-temporal dependency from inter-tower and in-tower traffic and predicts cellular traffic at a large scale.
[8]	-	DCRNN	Both	Learns a representation of the network and considers both the load on the links and the topological relations among them.
[129]	Cellular network	Gated Linear Unit, Graph Convolution	Both	Uses the transition probability matrix of the handover graph to improve traffic prediction.
[117]	Cellular network	GCN, GRU	Both	Captures Spatio-temporal features of the individual user traffic.
[105]	Cellular network	GNN, Attention mechanism	Both	Captures spatial-temporal cellular traffic.
[114]	Satellite network	GCN, GRU	Both	Demonstrates that the network topology, captured by GCN, could improve the network traffic prediction.

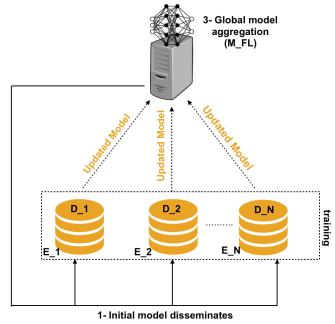


Fig. 3. General federated learning architecture.

the wireless traffic pattern and the geolocation information. Then, leveraging the augmented data collected from distributed BSs, a quasi-global prediction model has been constructed at the central server. This quasi-global model is used to mitigate the generalization difficulty of the global model caused by the statistical heterogeneity among traffic patterns collected from different clusters. Finally, instead of simply averaging the model weights collected from local clients to produce the global model, a dual attention-based model aggregation mechanism and a hierarchical aggregation structure are adopted at the central server. Using two real-world datasets, the authors compared *FedDA* against LSTM, SVM, and FedAvg models. In the same direction, Solat *et al.* [89] proposed a novel group management scheme by extending cluster FL as suggested in [120]. To reduce the average idle time and group formation cost of the edge servers, the genetic algorithm was used to optimize: i) the number of groups to be produced; and ii) the group association of the edge servers. Similarly, Kim *et al.* [51] suggested a joint approach to edge server selection and data set management to lower training costs, including training latency and energy consumption in FL for the NTP process by balancing accuracy and training cost. To achieve this, the sub-optimal solutions were found using a genetic algorithm. However, it can introduce additional complexity to the proposed solution. Using mutual information clustering, Zhang *et al.* [124] proposed an FL framework for wireless traffic prediction, called FedMIC. It employs a sliding window scheme, spectral clustering, and hierarchical aggregation architecture to improve client model learning, address non-independently and identically distributed data (non-IID), and reduce prediction error. Additionally, Perifanis *et al.* [72] used an FL approach to address several identified challenges due to the nature of non-IID data for traffic prediction. The experimental results show that FL achieves an equivalent prediction error to the centralized setting and can help minimize energy consumption and CO₂e emissions compared to centralized learning.

785 On the other hand, SDN and VNF are considered the enablers of the realization of the 5G
786 network. The SDN separates the data plane from the control plane in order to program the network
787 dynamically, and the VNFs are the software version of network functions deployed in virtual
788 environments. To better suit dynamic traffic load, SDN can be used to connect heterogeneous
789 distributed devices. In addition, NTP enables the SDN controller to react in advance to traffic
790 conditions by rerouting traffic to avoid future congestion. In this context, Sacco *et al.* [85] used the
791 LSTM model at the edge of the network to predict the future load and, in turn, optimize the routing
792 decision (i.e., select the best path). They used a federated architecture with a multi-agent control
793 plane (SDN controller), where each controller trains the LSTM model locally and sends only the
794 model parameters to the cloud. The results obtained confirm that this approach can reduce the
795 number of messages exchanged among the SDN controllers, speed up the training of the LSTM
796 model, and hence increase data delivery performance.

797 Furthermore, choosing the appropriate VNF instances is not a simple task. To solve this issue,
798 NTP can help autoscaling VNF based on expected traffic demand. Recently, Subramanya *et al.* [90]
799 proposed a traffic load prediction framework for auto-scaling of VNFs using an FL-based model. To
800 demonstrate the effectiveness of their approach, the authors evaluated the traffic prediction perfor-
801 mance using a Kubernetes-based orchestration prototype within a multi-access edge computing
802 platform. A comparison has been made against several centralized models such as LSTM, CNN, and
803 FFNN as well as they evaluate the performance of two FL approaches (i.e., with and without Model
804 Averaging). Zhang *et al.* [125] introduced a model-agnostic metalearning, MAML algorithm based on
805 the FL framework to achieve efficient mobile traffic prediction at the edge. They achieved this goal
806 by training an initial sensitive model that was highly adaptive to diverse mobile traffic statistics
807 in different locations. They used distance-based weighted model aggregation to their suggested
808 scheme and subsequently contrasted the results with several other conventional and FL-based algo-
809 rithms, FedAvg and FedDA [120]. Moreover, an intra-cluster FL-based model transfer framework for
810 mobile traffic prediction was presented by Li *et al.* [57]. The authors specifically used statistical and
811 geographical factors to group FL participants, and they discovered that manually defined features
812 worked well in this context. Through the use of DTL, edge servers with low processing power might
813 use models taught by edge servers with high processing power. It is noteworthy, however, that their
814 solution does not make use of other multidimensional data, including information on emergency
815 occurrences and area population density, in addition to some extra traffic parameters acquired by
816 AI. Furthermore, the work in [76] [75] a federated proximal long-short-term memory (FPLSTM)
817 framework by incorporating a proximal term to mitigate the impact of local models on the global
818 model. Next, after large-scale cellular networks received significant thought, the focus of this study
819 was broadened to include two different clustering techniques. In particular, random clustering and
820 information-based clustering using the geographic coordinates of the BSs and the traffic patterns
821 of the slices were implemented. These improvements helped increase robustness, scalability, and
822 precision. The NTP has significantly shifted from simple models to more advanced DL models and
823 then to FL. The majority of the works focused on improving the prediction error with DL and data
824 privacy concerns with FL; however, no study focuses on the sustainability of the proposed scheme.
825 In this context, Perifanis *et al.* [73] investigated the sustainability and predictive performance
826 of state-of-the-art DL models for federated cellular traffic forecasting. The authors introduced a
827 novel sustainability indicator to evaluate energy consumption for prediction error, which enables
828 convenient comparisons between various ML models in different experimental scenarios. The
829 results show that complex models have an enormous increase in energy consumption compared to
830 simpler models. Table 9 summarizes the used models, datasets, the extracted features as well as the
831 key contributions.

832
833

Table 9. Summary of Applications of FL-based Techniques in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[85]	Edge network	LSTM	Temporal	Chooses the best route according to the prediction results.
[120]	Cellular network	LSTM, K-means	Both	Mitigate the generalization difficulty of FedAvg caused by the statistical heterogeneity among collected traffic data.
[72]	5G network	MLP, RNN, LSTM, GRU, CNN	Temporal	Compares five machine learning models and extensive experimental studies using nine different aggregation algorithms, some of which are specifically designed for handling non-iid data.
[90]	5G network	CNN, LSTM, AE	Both	Performs the autoscaling of VNF instances based on the expected traffic demand and preserves the privacy of user data.
[76][75]	4G network	LSTM	Temporal	A federated proximal long short-term memory (FPLSTM) framework by incorporating a proximal term to mitigate the impact of local models on the global model.
[89]	Edge network	DL, clustering model	Both	Proposes a novel group management scheme based on clustered FL for mobile traffic prediction in mobile edge computing.
[51]	Edge network	Genetic algorithm, a concave estimation model	Temporal	Suggests a joint approach to edge server selection and data set management to lower training costs, including training latency and energy consumption in FL.
[124]	Wireless networks	Spectral clustering, LSTM	Both	Employs a sliding window scheme, spectral clustering, and hierarchical aggregation architecture to improve client model learning, address non-independently and identically distributed data.
[125]	Wireless network	Model-agnostic meta-learning	Both	Introduces a model-agnostic meta-learning, algorithm based on the FL framework to achieve efficient traffic prediction.
[57]	Core network	MLP	Both	Proposes an intra-cluster federated learning-based model transfer framework.
[73]	Cellular network	CNN, LSTM, Transformer,	Both	Addresses the trade-off between accuracy and energy consumption in FL by proposing a novel sustainability indicator that allows assessing the feasibility of DL models.

3.4 Existing benchmark

Using DL-based techniques and models for NTP is one of the key components of network management. However, among the DL models, no method outperforms all the others. These models have advantages and disadvantages depending on the application domain [9]. As a result, several benchmark papers have been proposed for link load prediction or TM prediction, using some open datasets. For example, Oliveira *et al.* [69] compared two neural network models, which are MLP and Stacked Autoencoder (SAE). The results demonstrate that SAE and MLP can accurately predict short-term network traffic. However, the SAE model reflects more computational complexity during training. Ramakrishnan *et al.* [81] proposed a comparative analysis of several RNN-based models, which are RNN, LSTM and GRU. Using two subsets (1000 flow measurements at each node) of public datasets (GÉANT and Abilene), they demonstrate that the LSTM and GRU models are more suitable than the RNN for network TM prediction problems. In addition, Aloraifan *et al.* [6] briefly compared the performance of different DL-based models for network TM prediction. In particular, the prediction performance of the single model (e.g., LSTM, GRU, and Bi-LSTM) and hybrid models (e.g., CNN+LSTM+Bi-LSTM and CNN+GRU+Bi-GRU) were compared. The experimental results on a subset of the GÉANT dataset show that hybrid models improve the prediction results as well, and using Bi-LSTM and Bi-GRU can add an extra level of assistance in capturing the temporal dependencies, and hence boost the prediction performance.

Similarly, Liu *et al.* [62] used different DL-based models for the overall prediction of TM and the prediction of the prediction flow of the load of the link separately. Using Abilene, CERNET, and GÉANT datasets, the experimental results show that predicting each link flow can improve the prediction accuracy, but it takes a longer time. Furthermore, Le *et al.* [53] compared the LSTM model with its two variants, the bidirectional LSTM (BiLSTM) and the GRU models for the prediction of link load in the SDN environment. Using two different datasets, the GÉANT dataset and a dataset generated by their network testbed, the GRU model demonstrates its prominence over LSTM and BiLSTM with a lower error rate.

Finally, since no model fits all the different traffic datasets, He *et al.* [40] proposed a metalearning scheme to predict adaptive and faster traffic. Specifically, the meta-learning scheme combines multiple predictors and consists of the master policy and several predictors called sub-policies. The master policy uses Deep Reinforcement Learning (DRL) to choose the best-fit predictor based on

recent prediction performance, whereas the subpolicies aim to predict specific traffic. The results of the experiment show that the proposed scheme outperforms single models and, in turn, could be an efficient solution for network traffic with different characteristics.

Although the proposed benchmark covered several DL-based models such as LSTM, GRU, and their variants (i.e. BiLSTM, BiGRU), no one considers GNN models. The GNN model helps to capture the spatial correlations in non-Euclidean data and in turn to compare its performance against other models, a comparative analysis could be a relevant direction. Table 10 summarizes the used models, datasets, the extracted features as well as the key contributions.

Table 10. Summary of benchmark of DL-based models in NTP.

Ref.	Network	Model	Spatial/Temporal	Contribution
[69]	Internet network	SAE, MLP	Temporal	Comparative analysis of MLP and Stacked Autoencoder (SAE).
[81]	-	RNN, LSTM, GRU	Temporal	Demonstrates high performance in traffic volume prediction for the different RNN-based models.
[6]	-	LSTM, GRU, Bi-LSTM, Bi-GRU, CNN	Both	A comparative analysis of single/simple model (e.g., LSTM, GRU, and Bi-LSTM) and hybrid models (e.g., CNN+LSTM+Bi-LSTM and CNN+GRU+Bi-GRU).
[53]	-	LSTM, Bi-LSTM, GRU	Temporal	Compares the performance of LSTM, Bi-LSTM, GRU for link load prediction in SDN environment.
[62]	-	GRU, LSTM	Temporal	Comparative analysis for overall TM prediction and the prediction of link load prediction flow separately.
[40]	4G network	LSTM	Temporal	Combines several models through a meta-learning scheme for adaptive and faster traffic prediction.

3.5 Lessons Learned

From the previous section, important remarks are made below.

- CNN and LSTM in Handling Spatiotemporal Data:** LSTM and CNN are prominently used for prediction tasks within network traffic analysis. Their frequent use underscores their effectiveness in handling spatial and temporal data characteristics, respectively. In particular, the hybrid integration of CNN and LSTM emerges as a highly effective strategy, providing significant enhancements in model performance by leveraging the strengths of both architectures to address complex prediction challenges. Recent advances, such as combining LSTMs with Transformers or incorporating attention mechanisms, can significantly enhance performance in scenarios requiring real-time adaptability and multi-scale dependency modeling.
- Enhancements Through Specialized Techniques:** Beyond combining DL models, various techniques have been employed to refine the prediction quality in terms of performance and privacy. These include TL, FL, MTL, Attention Mechanisms, and Graph Structure Information. Each technique addresses specific challenges, from using pre-learned patterns in TL to ensuring privacy and real-time responsiveness in FL, highlighting multifaceted approaches to improving DL models in network traffic prediction.
- Role of Transfer Learning in Tackling Data Scarcity:** TL is particularly valuable in scenarios with insufficient traffic data for training robust DL models. By applying knowledge from a related source domain, TL demonstrates promise, especially when the source and target domains share underlying patterns. This observation suggests a promising future research direction in which TL could be applied between similar domains, potentially unlocking new predictive capabilities.
- Privacy Preservation through Federated Learning (FL):** FL stands as a pivotal technique for addressing the increasing privacy concerns of service providers. Its decentralized nature allows models to be trained across distributed data sources without transferring raw data, which enhances both privacy and reduces bandwidth consumption. In the future, optimizing FL frameworks by integrating real-time federated learning and energy-efficient protocols will be crucial to managing massive and dynamic network traffic in a sustainable way.

- 932 • **Multi-task Learning for Resource Efficiency:** MTL has proven effective in reducing compu-
933 tational overhead by allowing a single model to simultaneously learn multiple tasks, such as
934 traffic prediction and anomaly detection. This approach provides a practical path toward scal-
935 able and efficient prediction systems, especially in high-demand networking environments.
936 Future research could explore integrating MTL with reinforcement learning or adaptive
937 learning techniques, where models can autonomously prioritize tasks based on network
938 conditions.
- 939 • **GNN for Improved Spatial Feature Extraction:** GNNs have shown superior ability to
940 extract spatial features compared to CNNs. This advantage points to the potential of GNNs
941 to improve model accuracy by better capturing the intricate spatial relationships inherent in
942 network traffic data.
- 943 • **The Promise of Transformers:** Transformers, with their self-attention mechanisms, have
944 revolutionized sequential data processing, enabling the modeling of long-range dependencies
945 and handling high-dimensional, real-time network traffic data more efficiently than tradi-
946 tional models like LSTMs and RNNs. Their ability to process sequences in parallel makes
947 them ideal for large-scale network environments where traffic flows dynamically. Recent
948 advancements, such as Large Language Models (LLMs) and Spatiotemporal Transformers,
949 further extend the potential of these models. LLMs, which excel at understanding complex
950 relationships in large datasets, can be fine-tuned to predict traffic surges or network con-
951 gestion by integrating contextual information such as user behavior and time-based events.
952 Spatiotemporal transformers go beyond time-series data, incorporating spatial relationships
953 between nodes in a network, which is essential for managing complex networks. These mod-
954 els can predict traffic across multiple locations by analyzing not only the temporal sequence
955 but also the spatial interdependence of traffic flows. This is particularly useful in scenarios
956 like 5G/6G networks, where real-time adaptability and low-latency predictions are crucial.
957 While transformers offer these significant advantages, their computational complexity is
958 a challenge. Hence, ongoing research into energy-efficient transformer architectures, such
959 as model pruning and attention sparsification, aims to reduce the computational burden,
960 making them viable for large-scale, real-time predictions with lower energy consumption.
- 961 • **Scalability and Real-time Adaptation:** With the explosive growth in network traffic and
962 its increasing complexity, scalability is a major concern. There is a pressing need for DL
963 models capable of processing streaming data in real-time while dynamically adapting to
964 network changes. This is especially crucial in modern applications like 5G/6G networks and
965 IoT traffic management, where latency is critical. Research in this direction should explore
966 scalable architectures, such as distributed transformers or asynchronous federated learning
967 models, that can handle large-scale, high-velocity data efficiently.
- 968 • **Energy Efficiency and Sustainability in DL Models:** The high computational demands
969 of DL models raise sustainability concerns, particularly regarding energy consumption. As
970 networks expand and predictive models become more complex, there is a need for energy-
971 efficient architectures that balance performance with sustainability. Techniques such as
972 model pruning, quantization, and neural architecture search (NAS) can help in designing
973 models that minimize energy use without compromising accuracy. The integration of green
974 AI principles into future research will be crucial to ensuring that network traffic prediction
975 models not only improve performance but also contribute to a lower environmental impact.
976
977
978
979
980

4 PUBLIC DATASETS

4.1 Studied datasets

For the sake of usage, the most commonly used datasets in network traffic prediction are presented below, and the references are listed in Table 11.

- **Abilene dataset:** The Abilene dataset contains trace data from the backbone network located in North America consisting of 12 nodes and 30 unidirectional links. The volume of traffic aggregated over 5-minute slots starting from March 1, 2004 to September 10, 2004 [1].
- **GEANT dataset:** The GEANT Topology has 23 nodes and 36 links. The TMs are summarized every 15 minutes starting from January 8th, 2005 for 16 weeks (10772 TMs) [98].
- **SDN dataset:** SDN network dataset is a recent new dataset for prediction tasks. These data were built through an SDN network testbed using a Mininet simulator. In the Mininet simulation, each node is represented by an OVS switch, and each switch is connected to a host that generates traffic flows in the network. The network consists of 14 nodes and 18 links with a temporal interval of 60 minutes. The proposed dataset contains 6257 TM after running for 4 days [53].
- **Telecom Italia dataset:** Telecom Italia is a part of the "Big Data challenge". It was collected from 01/11/2013 to 01/01/2014 with a temporal interval of 10 minutes over the whole city of Milan (62 days, 300 million records, about 19 GB). The area of Milan is divided into a grid of $H \times W$ (100×100) squares and the size of each square is about 235×235 meters and is referred to as a cell. In each cell, the service provider records three types of cell traffic: SMS, call service and Internet service [14].

Table 11. Public datasets for traffic prediction problems.

Dataset	Relevant studies
Abilene	[55] [35] [49] [8] [105] [68] [81] [62] [100]
GEANT	[12] [45] [35] [49] [68] [81] [6] [53] [62] [100]
Telecom Italia	[123] [47] [5] [87] [56] [105] [122] [28] [119] [108] [120] [61][44] [128]
SDN dataset	[53] [100]

4.2 Dataset analysis

In this section, we explore data dependency in both temporal and spatial domains of the different presented datasets.

• Analysis 1: Data Analysis in the Temporal Domain

The *sample autocorrelation function* (sample ACF) [18] is a widely used method to discover the data dependency in the temporal domain, which describes the dependency between the values of a sample process as a function of time lags h .

The definition of the ACF sample at cell / flow can be given as follows.

$$r_k = \frac{\sum_{t=1}^{T-h} (d_{t+h} - \bar{d})(d_t - \bar{d})}{\sum_{t=1}^T (d_t - \bar{d})^2}, 0 < k < T, \quad (9)$$

where T and \bar{d} are the total counts and mean value of data in the temporal dimension, respectively. The autocorrelation value lies in the range $[-1, 1]$. $r_k = 1$ indicates total positive autocorrelation between data with a time lag of h ; while $r_k = -1$ means total negative autocorrelation and $r_k = 0$ denotes no autocorrelation.

Fig. 4(a) 5(a) 6(a) 7(a) show the temporal behavior or the data. The x-axis denotes the time interval index, and the y-axis is the number of events of specific flow/cellular traffic. Fig. 4(c) 5(c) 6(c) 7(c) demonstrate that the traffic of the different datasets exhibits nonzero

autocorrelations in the time domain, and this indicates the future traffic volume can be predicted through historical observations. As the ACF value at lag 1 is close to 1, this indicates that there is a strong correlation between consecutive observations. This may be the case for data that has a strong trend or momentum behavior. For example, telecom Italia and SDN datasets have seasonal correlation because the ACF values show a repeating pattern of spikes at regular intervals, this indicates that there is a seasonal pattern in the data, which indicates that the data exhibit regular cycles or patterns over time.

- **Analysis 2: Data analysis in the spatial domain**

The spatial correlation of the traffic data is measured using a widely used metric [102], i.e. Pearson correlation coefficient ρ , between a target cell/flow (i, j) and its neighboring cells (i', j') , defined as follows.

$$\rho = \frac{\text{cov}(d_{i,j}, d_{i',j'})}{\sigma_{d_{i,j}} \sigma_{d_{i',j'}}} \quad (10)$$

where $\text{cov}()$ is the covariance operator, and σ is the standard deviation. Similarly, this ranges in $[-1, 1]$ as well.

The nonzero spatial correlation among neighboring cells/flows (Fig. 4(b) 5(b) 6(b) 7(b)) especially with GÉANT and telecom Italia datasets indicate clearly that the spatial correlation indeed exists among different cells/flows.

5 PERFORMANCE COMPARISON OF DL-BASED MODELS FOR NTP

In this section, we show the results of our performance comparison, which has been conducted on representative DL-based models such as LSTM, BiLSTM, GRU, BiGRU, Graph WaveNet (GWN) [110], and Diffusion Convolutional Recurrent Neural Networks model (DCRNN) [8]. DCRNN combines graph convolution networks with recurrent neural networks in an encoder-decoder manner, whereas GWN integrates diffusion graph convolutions with 1-D dilated convolutions.

We chose these models because they are among the most widely used for NTP, as well as to evaluate the performance of GNN-based models against the models used to extract the temporal correlations. To do so, we evaluated the performance of such models using three public traffic datasets, GÉANT, Abilene, and SDN datasets. Before explaining the construction of different models, it is important to process the data and take a look at it. Table 12 illustrates the characteristics of the datasets used and the figures. Fig. 4(a) 5(a) 6(a) presents the random selection flow pattern for 500 timestamps from the GÉANT, Abilene, and SDN datasets. Note that for a fair comparison of the patterns of the different datasets, we have used the same interval for the three datasets. In particular, we have aggregated the traffic of SDN and Abilene data to find the same interval of the GÉANT dataset (15 minutes).

We used two popular metrics: *MSE* and *MAE* (Equation 3 and Equation 5) as well as the inference time. In addition, during our experiments, we modeled TM prediction as a time series prediction problem. We assume that modeling each OD flow sequence separately may ignore the inherent correlations between OD flows as well as it requires longer training and predicting time.

5.1 Experiment Setup

The models are implemented using *Python 3* as a programming language and *PyTorch* as a DL framework. The datasets have been divided chronologically into a training set, a validation set, and a test set according to the proportions of 70%, 10%, and 20%, respectively. Table 12 illustrates the characteristics of the data sets used and the figures. Then, a normalization of each traffic flow was

1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127

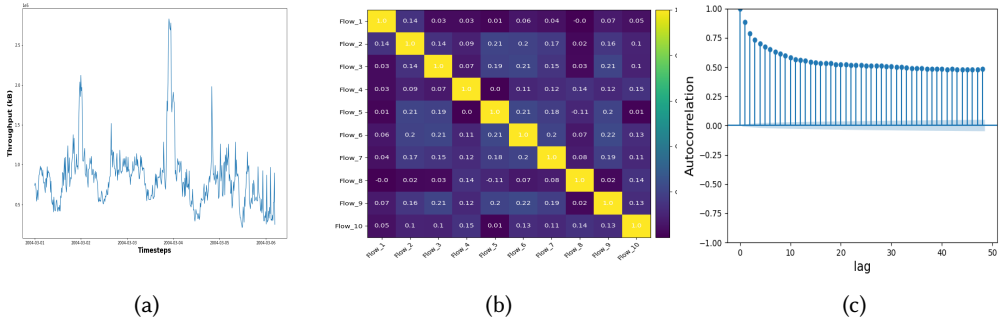


Fig. 4. The spatial and temporal dynamics of the Abilene dataset. (a) Temporal dynamic of the traffic; (b) Spatial correlation; (c) Autocorrelation analysis

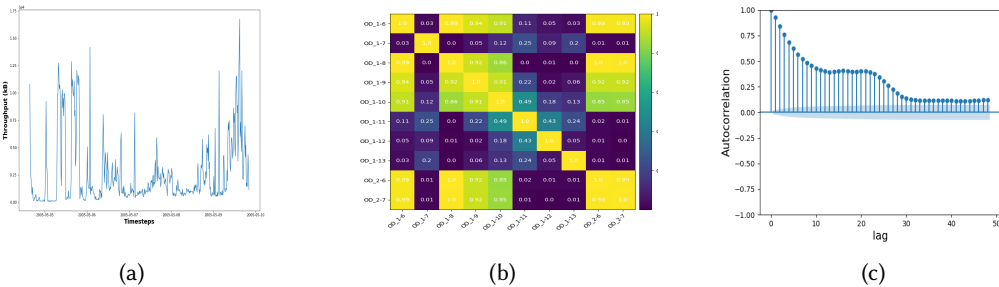


Fig. 5. The spatial and temporal dynamics of the GÉANT dataset. (a) Temporal dynamic of the traffic; (b) Spatial correlation; (c) Autocorrelation analysis

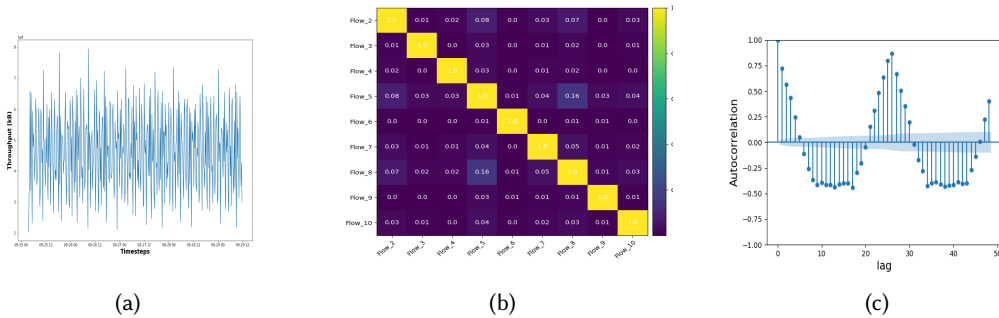


Fig. 6. The spatial and temporal dynamics of the SDN dataset. (a) Temporal dynamic of the traffic; (b) Spatial correlation; (c) Autocorrelation analysis

performed, where all values are in the range of $[0, 1]$ to optimize the performance of the training process. The code is available online: https://github.com/aouedions11/Network_Traffic_prediction.

Table 12. Datasets description

Dataset	Nodes	Flows	Interval	Records
GÉANT	23	529	15 min	10,769
Abilene	12	144	5 min	48,096
SDN	14	196	1 min	6,257

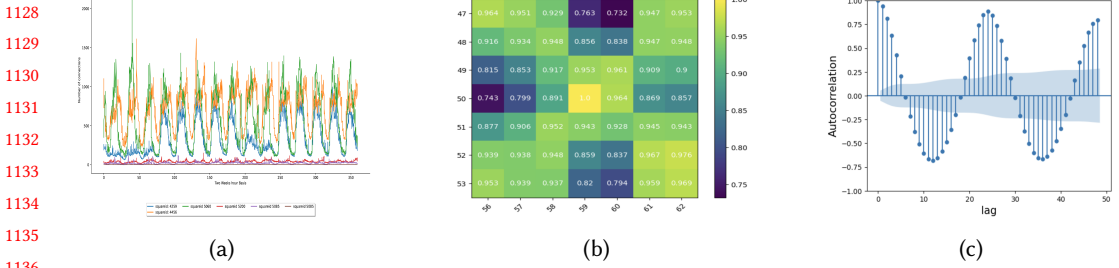


Fig. 7. The spatial and temporal dynamics of the Telecom Italia dataset. (a) Temporal dynamic of the traffic; (b) Spatial correlation; (c) Autocorrelation analysis

5.2 Prediction performance

In this first set of experiments, the different models have been trained for 500 epochs with sequence length set to 24 with GÉANT and Abilene datasets and 60 with SDN datasets. Only three datasets were chosen instead of the four mentioned above for several reasons. First, the three selected datasets all represent backbone network data. Backbone networks form the main Internet pathways, and understanding their behavior is crucial for ensuring effective and reliable Internet communication. The Internet network facilitate the bulk of data transmission, and their performance directly influences the end-user experience. By focusing on these datasets, our work can identify patterns that would directly affect internet traffic. Second, the three datasets vary in network size and interval, allowing for a more comprehensive analysis and robust findings that can be generalized across various network contexts. Thus, we focus our analysis on the backbone network datasets to maintain consistency in the study and ensure that the results are directly relevant to the domain of interest. The results are illustrated in Table 13.

Table 13. MSE, MAE ($\times 10^{-3}$), and inference time of the different DL-based models on GÉANT, Abilene, and SDN datasets

Metric	GÉANT Dataset			Abilene dataset			SDN dataset		
	MSE	MAE	Inference Time	MSE	MAE	Inference Time	MSE	MAE	Inference Time
LSTM	1.644	13.159	0.278	7.529	29.225	0.969	13.349	67.071	0.306
BiLSTM	1.294	11.662	0.482	6.289	22.006	1.711	13.123	66.189	0.574
GRU	1.592	12.907	0.248	7.354	28.472	0.799	12.686	64.891	0.258
BiGRU	1.244	11.559	0.417	6.188	21.416	1.458	12.477	64.776	0.48
GWN	0.879	5.954	3.651	6.220	18.318	3.814	7.936	52.927	2.694
DCRNN	4.166	26.430	8.91	14.507	59.543	32.729	67.892	213.923	9.845

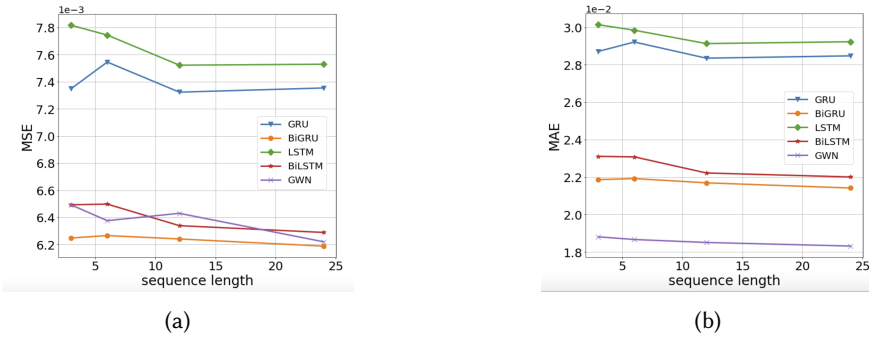
As can be seen, the GWN model gives the best results, followed by BiGRU, BiLSTM, LSTM, and DCRNN. The performance of GWN is mainly due to its ability to capture the spatial relation in the TMs, which is not the case for the other models. Moreover, there is one exception where GRU performs slightly better than the BiLSTM model on the SDN dataset. This exception may be attributed to the fact that the SDN dataset does not require complex models like BiLSTM, and this also can be seen with the results of DCRNN. Furthermore, the seasonality of the SDN dataset (Fig. 6 (c)) helps the model to better capture the temporal dependencies.

On the other hand, with all datasets, the DCRNN shows the worst results in terms of MAE and MSE. This is because SDN contains sudden high peaks more than the GÉANT and Abilene datasets (as shown in Fig. 6 (a)) where DCRNN hardly predicts such events [8] since it was initially proposed for intelligent transport systems that have fewer fluctuations than network traffic. In addition, the DCRNN model is the most time consuming for inference among the other models.

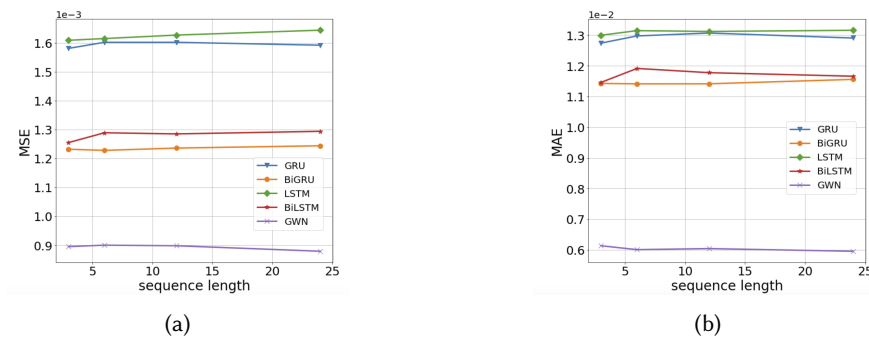
1177 From these results, though GWN gives the best results, we can notice also that simple models can
 1178 perform better than graph-based models (DCRNN), and thus the models' performance depends on
 1179 the dataset and the context. Furthermore, we can notice that DCRNN and GWN perform better
 1180 with the GÉANT dataset than with the Abilene and SDN datasets. This advantage will become even
 1181 more significant in comparison to the SDN dataset. This is mainly due to the high spatial correlation
 1182 among the flow in the GÉANT dataset (Fig. 5(c)). Consequently, these results demonstrate that
 1183 GNN-based models work well and can provide good performance with highly correlated traffic.
 1184

1185 **5.3 Impact of sequence length**

1186 In this subsection, we study the relationship between the performance of the DL-based models and
 1187 the sequence-length hyperparameters. Fig. 8, Fig. 9, Fig. 10 and show the impact of the sequence
 1188 length on the data sets *MSE* and *MAE* on the Abilene, GÉANT, and SDN datasets. Different sequence
 1189 length is used as the input to the models to predict the next single TM volume. It can be seen that
 1190 in most cases the performance of all models increases as the sequence length increases. Therefore,
 1191 we can believe the fact that these models can remember what they have learned before many time
 1192 steps. Additionally, this may be attributed to the fact that longer sequence lengths increase the size
 1193 of the training set and in turn improve the generalization capability of the models. At the same
 1194 time, we can see that the GWN model maintained a better prediction performance than the other
 1195 models. With the support of stacked dilated casual convolutions, GWN can handle spatial-temporal
 1196 graph data with long-range temporal sequences efficiently.
 1197



1208 Fig. 8. Impact of sequence length on the prediction performance of different models on Abilene dataset



1222 Fig. 9. Impact of sequence length on the prediction performance of different models in the GÉANT dataset

1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274

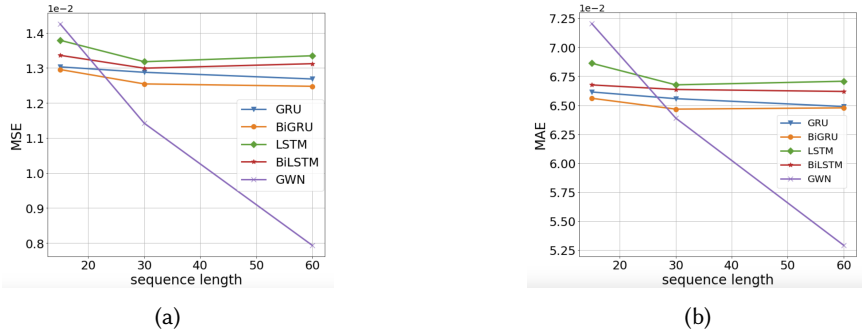


Fig. 10. Impact of sequence length on the prediction performance of different models in SDN dataset

5.4 Flow-by-Flow prediction vs. TM prediction

In this section, we compare the prediction performance of flow-by-flow prediction and TM prediction methods. To evaluate these methods, we used two well-known models, LSTM and GRU on the two datasets (because Abilene is very expensive for flow-by-flow cases). For flow-by-flow prediction, we assumed that each OD traffic is independent of all other ODs, and we fed the LSTM/GRU model with flow-by-flow. In particular, with the flow-by-flow TM prediction method, the model uses the traffic volumes of the OD flows in the TM to predict one by one.

As shown in Fig. 11 and Fig. 12, using the LSTM model, the TM prediction outperforms the flow-by-flow prediction models. This is due to the inter-flow correlation that may help to reduce the error, whereas flow-by-flow prediction ignores the relationship between the nodes and the importance of correlations among network traffic flows.

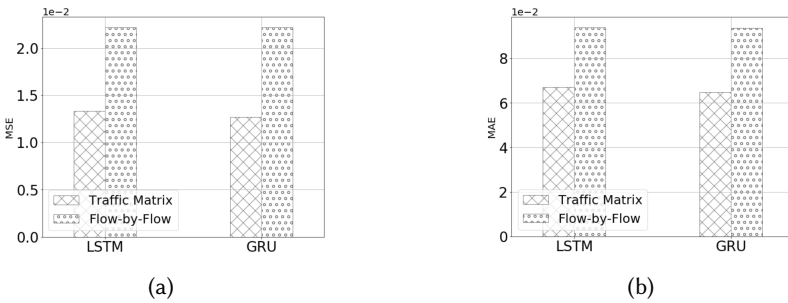


Fig. 11. Performance comparison of TM vs. flow-by-flow prediction with LSTM and GRU on SDN dataset

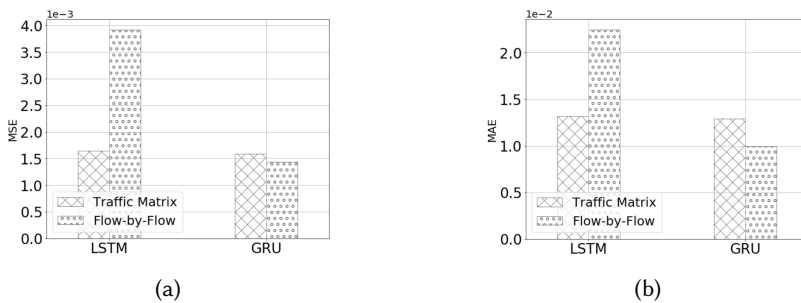


Fig. 12. Performance comparison of TM vs. flow-by-flow prediction with LSTM and GRU on GÉANT dataset

6 OPEN ISSUES AND FUTURE DIRECTIONS

The application of DL models for NTP has yielded promising results. However, several critical challenges remain in designing and implementing these solutions in practical settings [19]. These challenges encompass the selection of appropriate DL architectures, explainability, energy efficiency, and the integration of traffic prediction with broader network management tasks. Furthermore, new technologies and methods, such as LLMs and Physics-Informed Neural Networks (PINNs), are emerging as important considerations for the future. In this section, we outline key challenges and propose research opportunities to address them. These challenges, along with future directions, are summarized in Table 14.

6.1 Identifying DL architecture

Selecting an appropriate DL architecture for NTP remains a significant challenge. While models like LSTM, CNN, and Transformers have demonstrated strong predictive performance, determining the optimal architecture is often a complex task that requires tuning many hyperparameters, which can grow exponentially with model depth. Additionally, the choice of architecture depends heavily on the nature of the data, computational resources, and the specific application. In recent years, reinforcement learning and genetic algorithms have shown promise in automating the hyperparameter tuning process. For example, Q-learning can help systematically select the most appropriate architecture by evaluating model performance based on a reward system. Future research should focus on leveraging these techniques to not only improve model accuracy but also reduce the computational burden associated with model selection.

6.2 Explainable DL models

Due to the black-box nature of DL-based models, it is very difficult for network operators to understand the reasons behind their outputs, causing a lack of trust in such models [26]. The design of an DL model for traffic prediction that provides interpretable decisions about the final result is of paramount importance to gain the trust of the end user. The challenges above can be solved with eXplainable AI (XAI). To overcome this, Explainable AI (XAI) techniques are essential. XAI can provide insights into how a model arrives at its predictions, improving user trust and enabling operators to make informed decisions [38]. As a result, several XAI techniques have been proposed, but the majority of them treat static data [84]. Contrarily to static data, temporal data for traffic forecasting are complex, they may be non-stationary. As a result, several methods have been proposed. In particular, the framework proposed in [30] can be used to evaluate and benchmark the performance-explainability characteristics of various models, including those designed for traffic prediction based on DL models. On the other hand, to explain the CNN model applied to time series, the authors in [22] proposed a 'Gradient*Input' method that computes the partial derivative of the current layer for the input and multiplies it by the input itself. Therefore, they computed neurons and filter activation for a specific instance. The input subsequences processed by the most activated filters have the highest contribution to the prediction. In addition, perturbation techniques, such as that in [84], involve removing or masking certain input features to determine their contribution to the final prediction [7]. Moreover, attention mechanisms can enhance interpretability by highlighting which portions of the input data the model focused on. For example, the work in [23] combines a CNN as a feature extractor and a LSTM model to learn the temporal dependencies. Then, the hidden states and output states of the LSTM are used as input of a feedforward neural network layer that performs classification. The weights of this feedforward layer are the attention weights that indicate the importance of the different timesteps of the time series. Further research should

1324 explore the development of explainable spatiotemporal models, particularly in complex NTP tasks
1325 involving non-stationary data.

1326 **6.3 Leveraging LLMs for NTP**

1327 The introduction of LLMs like GPT and BERT has revolutionized sequence modeling, and these
1328 advancements can be adapted for NTP. LLMs can capture intricate, long-range dependencies in
1329 data, allowing for more context-aware and accurate predictions. LLMs could significantly enhance
1330 the predictive power of NTP models by integrating external factors, such as weather conditions or
1331 user feedback, into the prediction framework [43]. Moreover, their ability to generalize across tasks
1332 makes them well-suited for cross-domain applications, such as transferring predictive knowledge
1333 between different network environments. Future work should explore fine-tuning LLMs specifically
1334 for NTP, addressing the scalability challenges posed by the high-dimensional nature of traffic data.

1335 **6.4 DL enabled green networking technology**

1336 As DL-based models require massive computations to achieve acceptable performance, for exam-
1337 ple, training relatively simple CNN models still requires several CPU cycles and huge memory
1338 space. Recently, great attention has been paid to energy consumption during model training and
1339 inference [113]. In this context, to reduce the computation cost of DL-based approaches, several
1340 mechanisms can be explored such as a lightweight model and using fewer data to train our model
1341 while keeping the model performance. Using lightweight ML/DL models is helpful to get faster
1342 prediction and to achieve the trade-off between energy consumption and model performance.
1343 Several possible lightweight models should be considered for traffic prediction. One direction is to
1344 use Binary Neural Network (BNN) [78] requires less memory and computational resources due to
1345 the binary values for both activations and weights, simplifying computation, and making them
1346 suitable for low-power devices and resources-constrained. Furthermore, Spiking Neural Networks
1347 (SNNs), which are next-generation artificial neural networks inspired by information processing
1348 in biology. SNNs can reduce computational complexity while maintaining competitive predictive
1349 accuracy, making them ideal for resource-constrained environments like edge computing [96].
1350 In addition, reducing the model size is an essential practical problem in such network-related
1351 scenarios, and Knowledge Distillation (KD) can be a useful technique for this problem. KD is a
1352 technique that enables the transfer of knowledge from a large network that has been trained to
1353 solve a certain task to a smaller network [41].
1354

1355 **6.5 Integrating Traffic Prediction with Resource Management**

1356 NTP and resource management are usually performed independently, while traffic prediction can
1357 play an important role in network management, such as short- and long-term resource allocation,
1358 anomaly detection, and traffic routing. Integrating traffic prediction with resource management
1359 offers numerous potential benefits for wireless networks. Using predictive analytics, network
1360 operators can anticipate future traffic trends and proactively allocate resources to meet demand,
1361 thus improving network efficiency and user satisfaction. Furthermore, integration can enable
1362 more effective load balancing, congestion management, and fault detection, leading to a more
1363 robust and resilient network infrastructure. A limitation of separately modeling NTP and resource
1364 management is that the potential relations between traffic prediction and resource allocation are
1365 ignored, which can degrade the performance of QoS / QoE. Consequently, it is interesting to
1366 have a model that jointly solves different but related problems. In this context, [54] solves traffic
1367 engineering problems based on segment routing by taking into account future traffic changes. The
1368 results of the experiment show that the proposed solution can achieve near-optimal performance
1369 in terms of maximum link utilization and significantly reducing the number of routing changes.
1370 Furthermore, connected users can be moved and connected from one base station to another base
1371

1373 station, with the transfer process. In this context, the use of the transition probability matrix of the
1374 handover graph can improve traffic prediction [129].

1375 6.6 Physics-Informed Neural Networks (PINNs) for NTP

1377 Traditional DL models applied to NTP often overlook the underlying physical principles that
1378 govern network traffic flows, such as conservation laws and network topology constraints. Physics-
1379 Informed Neural Networks (PINNs) present a novel approach by embedding physical knowledge
1380 directly into the learning process [24]. This could lead to more accurate and interpretable predictions,
1381 as the models would adhere to known constraints of network behavior. Incorporating PINNs into
1382 NTP frameworks could also improve model generalization, as these models would be less likely to
1383 produce unrealistic predictions under novel conditions. Research in this area should explore how
1384 PINNs can be adapted to model the highly dynamic and nonlinear nature of network traffic.

1385 6.7 Concept Drift-aware DL for NTP

1387 Concept drift in NTP could arise from changes in network configurations, user behaviors, or network
1388 conditions. Therefore, it is essential to continuously adapt the predictive models to these changes.
1389 Consequently, continuous adaptation of predictive models to maintain accuracy and relevance is the
1390 need of the hour. Integrating concept drift-aware mechanisms into existing DL architectures without
1391 introducing excessive computational overhead or sacrificing predictive performance remains a
1392 key challenge. To do so, one should continuously update the DL models to accommodate the drift.
1393 Although continuous updates in the DFL for NTP can offer better performance, this will be at
1394 the cost of energy consumption. In this context, Deep Unlearning (DUL) [79] can be used to find
1395 the tradeoff between performance and energy consumption. DUL can selectively discard outdated
1396 or irrelevant knowledge while retaining valuable information relevant to the current state of the
1397 network. In addition, using KD techniques helps transfer knowledge from the original model to an
1398 updated version trained on recent data [37]. By distilling the knowledge learned from past network
1399 traffic patterns into a more compact and up-to-date model, incremental learning can facilitate
1400 seamless adaptation to concept drift while minimizing computational overhead.

1401 7 CONCLUSION

1403 In recent years, numerous studies and efforts have been made to predict traffic. In this paper, we
1404 examine the use of DL for NTP, a promising topic that enables intelligent network management,
1405 since accurate network traffic prediction reduces unnecessary resources by tightly allocating
1406 network, cache, and computing resources based on future traffic demand. First, we provide a brief
1407 introduction to DL and NTP. There is also a detailed discussion of papers addressing such a problem
1408 in terms of the types of DNN techniques used, such as simple/hybrid models, transformers, transfer
1409 learning, multitask learning, federated learning, and graph neural networks. Additionally, we
1410 provide a list of publicly available datasets and their behavior, as well as present some open-source
1411 projects for future research. Moreover, to compare and study the performance of some models
1412 under different scenarios, we conducted numerical experiments on some well-used datasets. Finally,
1413 we discuss some of the major challenges and directions for future research.

1414 In future studies, we can expand our findings by incorporating a wider spectrum of datasets,
1415 including those such as Telecom Italia, to offer a more comprehensive view of network behavior
1416 across different infrastructures. Moreover, the exploration of advanced DL architectures, especially
1417 Transformer models, stands to open new avenues in NTP, given their profound capabilities in
1418 capturing intricate patterns. Such advances could lead to better predictive accuracy and an enhanced
1419 ability to adapt to the dynamism of network traffic. Furthermore, future research should investigate
1420 online learning and self-supervised learning for NTP. These approaches have the potential to
1421

Table 14. Summary of Challenges and Future Directions For DL for Network Traffic Prediction

Challenges	Description	Future Directions
Identifying DL architecture	Finding suitable architecture and identifying optimal hyperparameters are difficult tasks and can influence the model performance.	1. Using reinforcement learning or a genetic algorithm for hyperparameter tuning for DL-based models [83] [3]
Explainable DL models	Lack of justification/ interpretability of the final prediction	1. Perturbation techniques such as removing or masking certain input features to determine their contribution to the final prediction [7]. 2. Attention mechanisms can be utilized to address the interpretability of the prediction [23] as they assign values that correspond to the importance of different parts of the time series according to the model.
DL-enabled green network technology	Reducing the energy consumption during model training and inference	1. A lightweight ML/DL model is helpful to get faster predictions and to achieve the trade-off between energy consumption such as spiking neural networks [101] and binary neural networks [78]. 2. Knowledge distillation is a useful technique for energy consumption that enables the transfer of knowledge from a large network that has been trained to solve a certain task to a smaller network [41].
Fuse Traffic prediction with resource management	Improve resource management through traffic prediction and vice versa	1. Using the transition probability matrix of the handover graph captures the spatial characteristics of the traffic and in turn, improves the prediction [129]. 2. Using the predicted results for traffic engineering including maximum link utilization [54]
Concept Drift-aware DL	Discard outdated or irrelevant knowledge while retaining valuable information relevant to the current network state.	1. Using Deep Unlearning [79] to find the tradeoff between performance and energy consumption. 2. Using Knowledge Distillation Techniques to Transfer Knowledge from the Original Model to an Updated Version Training on Recent Data [37].
Leveraging LLMs for NTP	LLMs, such as GPT and BERT, offer the ability to capture long-range dependencies and integrate external context, enhancing NTP performance.	1. Fine-tuning LLMs for NTP tasks and addressing the challenge of scaling LLMs to handle high-dimensional traffic data [43]. 2. Exploring cross-domain applications where LLMs transfer predictive knowledge across different network environments.
Physics-Informed Neural Networks (PINNs) for NTP	Traditional DL models often ignore underlying physical principles like conservation laws and network topologies. PINNs incorporate this knowledge into the learning process.	1. Adapting PINNs to account for the complex, nonlinear nature of network traffic while embedding physical constraints directly into the model [24].

significantly enhance the adaptability and efficiency of predictive models by allowing them to learn from new data in real time and extract valuable insights from unlabeled data, respectively.

REFERENCES

- [1] [n. d.]. . <http://www.cs.utexas.edu/~yzhang/research/AbileneTM/>.
- [2] Mahmoud Abbasi, Amin Shahraki, and Amir Taherkordi. 2021. Deep learning for network traffic monitoring and analysis (NTMA): A survey. *Computer Communications* (2021).
- [3] Shaashwat Agrawal, Sagnik Sarkar, Mamoun Alazab, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, Quoc-Viet Pham, et al. 2021. Genetic CFL: hyperparameter optimization in clustered federated learning. *Computational Intelligence and Neuroscience* 2021 (2021).
- [4] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. 2010. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews* 29, 5-6 (2010), 594-621.
- [5] Imad Alawe, Adlen Ksentini, Yassine Hadjadj-Aoul, and Philippe Bertin. 2018. Improving traffic forecasting for 5G core network scalability: A machine learning approach. *IEEE Network* 32, 6 (2018), 42-49.
- [6] Dalal Aloraifan, Intiaz Ahmad, and Ebrahim Alrashed. 2021. Deep learning based network traffic matrix prediction. *International Journal of Intelligent Networks* 2 (2021), 46-56.
- [7] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104* (2017).
- [8] Davide Andreoletti, Sebastian Troia, Francesco Musumeci, Silvia Giordano, Guido Maier, and Massimo Tornatore. 2019. Network traffic prediction based on diffusion convolutional recurrent neural networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 246-251.
- [9] Ons Aouedi, Kandaraj Piamrat, and Benoît Parrein. 2021. Decision tree-based blending method using deep-learning for network management. In *IEEE/IFIP Network Operations and Management Symposium*.
- [10] Ons Aouedi, Kandaraj Piamrat, and Benoît Parrein. 2022. Intelligent Traffic Management in Next-Generation Networks. *Future internet* 14, 2 (2022), 44.
- [11] Amin Azari, Panagiotis Papapetrou, Stojan Denic, and Gunnar Peters. 2019. Cellular traffic prediction and classification: A comparative evaluation of LSTM and ARIMA. In *International Conference on Discovery Science*. Springer, 129-144.

- 1471 [12] Abdelhadi Azzouni and Guy Pujolle. 2018. "NeuTM: A neural network-based framework for traffic matrix prediction
1472 in SDN". In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*. Taipei, Taiwan,
1473 1–5.
- 1474 [13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to
1475 align and translate. *arXiv preprint arXiv:1409.0473* (2014).
- 1476 [14] Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio
1477 Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. 2015. A multi-source dataset of urban life in the
1478 city of Milan and the Province of Trentino. *Scientific data* 2, 1 (2015), 1–15.
- 1479 [15] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. "Representation learning: A review and new perspectives".
1480 *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
- 1481 [16] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. 2012. Machine learning strategies for time series
1482 forecasting. In *European business intelligence summer school*. Springer, 62–77.
- 1483 [17] Raouf Boutaba, Mohammad A Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and
1484 Oscar M Caicedo. 2018. A comprehensive survey on machine learning for networking: evolution, applications and
1485 research opportunities. *Journal of Internet Services and Applications* 9, 1 (2018), 1–99.
- 1486 [18] Peter J Brockwell and Richard A Davis. 2002. *Introduction to time series and forecasting*. Springer.
- 1487 [19] Pedro Casas. 2020. Two Decades of AI4NETS-AI/ML for Data Networks: Challenges & Research Directions. In *NOMS*
1488 *2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–6.
- 1489 [20] Zekai Chen, E Jiaze, Xiao Zhang, Hao Sheng, and Xiuzheng Cheng. 2020. Multi-task time series forecasting with
1490 shared attention. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 917–925.
- 1491 [21] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and
1492 Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation.
1493 *arXiv preprint arXiv:1406.1078* (2014).
- 1494 [22] Sohee Cho, Ginkyeng Lee, Wonjoon Chang, and Jaesik Choi. 2020. Interpretation of deep temporal representations
1495 by selective visualization of internally activated nodes. *arXiv preprint arXiv:2004.12538* (2020).
- 1496 [23] Kyu Sung Choi, Seung Hong Choi, and Bumseok Jeong. 2019. Prediction of IDH genotype in gliomas with dynamic
1497 susceptibility contrast perfusion MR imaging using an explainable recurrent neural network. *Neuro-oncology* 21, 9
1498 (2019), 1197–1209.
- 1499 [24] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli.
1500 2022. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal*
1501 *of Scientific Computing* 92, 3 (2022), 88.
- 1502 [25] Fatoumata Dama and Christine Sinoquet. 2021. Analysis and modeling to forecast in time series: a systematic review.
1503 *arXiv preprint arXiv:2104.00164* (2021).
- 1504 [26] Arun Das and Paul Rad. 2020. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv*
1505 *preprint arXiv:2006.11371* (2020).
- 1506 [27] Jan G De Gooijer and Rob J Hyndman. 2006. 25 years of time series forecasting. *International journal of forecasting* 22,
1507 3 (2006), 443–473.
- 1508 [28] Aicha Dridi, Hossam Afifi, Hassine MOUNGLA, and Chérifa Boucetta. 2021. Transfer learning for classification and
1509 prediction of time series for next generation networks. In *ICC 2021-IEEE International Conference on Communications*.
1510 IEEE, 1–6.
- 1511 [29] Alessandro D'Alconzo, Idilio Drago, Andrea Morichetta, Marco Mellia, and Pedro Casas. 2019. A survey on big data
1512 for network traffic monitoring and analysis. *IEEE Transactions on Network and Service Management* 16, 3 (2019),
1513 800–813.
- 1514 [30] Kevin Fauvel, Véronique Masson, and Elisa Fromont. 2020. A performance-explainability framework to benchmark
1515 machine learning methods: application to multivariate time series classifiers. *arXiv preprint arXiv:2005.14501* (2020).
- 1516 [31] Jie Feng, Xinlei Chen, Rundong Gao, Ming Zeng, and Yong Li. 2018. Deeptp: An end-to-end neural network for
1517 mobile cellular traffic prediction. *IEEE Network* 32, 6 (2018), 108–115.
- 1518 [32] Gabriel O Ferreira, Chiara Ravazzi, Fabrizio Dabbene, Giuseppe C Calafiore, and Marco Fiore. 2023. Forecasting
1519 Network Traffic: A Survey and Tutorial with Open-Source Comparative Evaluation. *IEEE Access* (2023).
- [33] Alex M Fout. 2017. *Protein interface prediction using graph convolutional networks*. Ph.D. Dissertation. Colorado State
University.
- [34] Kaihui Gao, Dan Li, Li Chen, Jinkun Geng, Fei Gui, Yang Cheng, and Yue Gu. 2020. Incorporating intra-flow
dependencies and inter-flow correlations for traffic matrix prediction. In *2020 IEEE/ACM 28th International Symposium*
on Quality of Service (IWQoS). IEEE, 1–10.
- [35] Kaihui Gao, Dan Li, Li Chen, Jinkun Geng, Fei Gui, Yang Cheng, and Yue Gu. 2020. Predicting Traffic Demand
Matrix by Considering Inter-flow Correlations. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*
Workshops (INFOCOM WKSHPs). IEEE, 165–170.

- 1520 [36] Yin Gao, Man Zhang, Jiajun Chen, Jiren Han, Dapeng Li, and Ruitao Qiu. 2021. Accurate load prediction algorithms as-
 1521 sisted with machine learning for network traffic. In *2021 International Wireless Communications and Mobile Computing*
 1522 *(IWCMC)*. IEEE, 1683–1688.
- 1523 [37] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International*
 1524 *Journal of Computer Vision* 129 (2021), 1789–1819.
- 1525 [38] David Gunning. 2017. Explainable artificial intelligence (xai). *Defense advanced research projects agency (DARPA), nd*
 1526 *Web 2*, 2 (2017), 1.
- 1527 [39] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu,
 1528 Yixing Xu, et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*
 1529 45, 1 (2022), 87–110.
- 1530 [40] Qing He, Arash Moayyedi, György Dán, Georgios P Koudouridis, and Per Tengkvist. 2020. A meta-learning scheme
 1531 for adaptive short-term network traffic prediction. *IEEE Journal on Selected Areas in Communications* 38, 10 (2020),
 1532 2271–2283.
- 1533 [41] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint*
 1534 *arXiv:1503.02531* (2015).
- 1535 [42] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- 1536 [43] Chengming Hu, Hao Zhou, Di Wu, Xi Chen, Jun Yan, and Xue Liu. 2024. Self-Refined Generative Foundation Models
 1537 for Wireless Traffic Prediction. *arXiv preprint arXiv:2408.10390* (2024).
- 1538 [44] Yahui Hu, Yujiang Zhou, Junping Song, Luyang Xu, and Xu Zhou. 2022. Citywide Mobile Traffic Forecasting
 1539 Using Spatial-Temporal Downsampling Transformer Neural Networks. *IEEE Transactions on Network and Service*
 1540 *Management* 20, 1 (2022), 152–165.
- 1541 [45] Yuxiu Hua, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu, and Honggang Zhang. 2019. Deep learning with
 1542 long short-term memory for time series prediction. *IEEE Communications Magazine* 57, 6 (2019), 114–119.
- 1543 [46] Chih-Wei Huang, Chiu-Ti Chiang, and Qiuhui Li. 2017. A study of deep learning networks on mobile traffic forecasting.
 1544 In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*.
 1545 IEEE, 1–6.
- 1546 [47] Shan Jaffry. 2020. Cellular traffic prediction with recurrent neural network. *arXiv preprint arXiv:2003.02807* (2020).
- 1547 [48] Weiwei Jiang. 2022. Cellular traffic prediction with machine learning: A survey. *Expert Systems with Applications*
 1548 (2022), 117163.
- 1549 [49] Weiwei Jiang. 2022. Internet traffic matrix prediction with convolutional LSTM neural network. *Internet Technology*
 1550 *Letters* 5, 2 (2022), e322.
- 1551 [50] Maryam Khalid. 2023. Traffic Prediction in Cellular Networks using Graph Neural Networks. *arXiv preprint*
 1552 *arXiv:2301.12605* (2023).
- 1553 [51] Doyeon Kim, Seungjae Shin, Jaewon Jeong, and Joohyung Lee. 2023. Joint Edge Server Selection and Dataset
 1554 Management for Federated Learning-enabled Mobile Traffic Prediction. *IEEE Internet of Things Journal* (2023).
- 1555 [52] Pedro Lara-Benitez, Manuel Carranza-Garcia, and José C Riquelme. 2021. An experimental review on deep learning
 1556 architectures for time series forecasting. *International Journal of Neural Systems* 31, 03 (2021), 2130001.
- 1557 [53] Duc-Huy Le, Hai-Anh Tran, Sami Souihi, and Abdelhamid Mellouk. 2021. An AI-based Traffic Matrix Prediction
 1558 Solution for Software-Defined Network. In *ICC 2021-IEEE International Conference on Communications*. IEEE, 1–6.
- 1559 [54] Van An Le, Tien Thanh Le, Phi Le Nguyen, Huynh Thi Thanh Binh, and Yusheng Ji. 2021. Multi-time-step Segment
 1560 Routing based Traffic Engineering Leveraging Traffic Prediction. In *2021 IFIP/IEEE International Symposium on*
 1561 *Integrated Network Management (IM), Bordeaux, France*. 125–133.
- 1562 [55] Van An Le, Phi Le Nguyen, and Yusheng Ji. 2019. Deep Convolutional LSTM Network-based Traffic Matrix Prediction
 1563 with Partial Information. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington,*
 1564 *VA, USA*. 261–269.
- 1565 [56] Ming Li, Yuewen Wang, Zhaowen Wang, and Huiying Zheng. 2020. A deep learning method based on an attention
 1566 mechanism for wireless network traffic prediction. *Ad Hoc Networks* 107 (2020), 102258.
- 1567 [57] Pengyu Li, Yingji Shi, Yanxia Xing, Chaorui Liao, Menghan Yu, Chengwei Guo, and Lei Feng. 2022. Intra-Cluster
 1568 Federated Learning-Based Model Transfer Framework for Traffic Prediction in Core Network. *Electronics* 11, 22
 1569 (2022), 3793.
- 1570 [58] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven
 1571 traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).
- 1572 [59] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven
 1573 traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).
- 1574 [60] Lei Liu, Yuxing Tian, Chinmay Chakraborty, Jie Feng, Qingqi Pei, Li Zhen, and Keping Yu. 2023. Multilevel Federated
 1575 Learning based Intelligent Traffic Flow Forecasting for Transportation Network Management. *IEEE Transactions on*
 1576 *Network and Service Management* (2023).

- 1569 [61] Qingyao Liu, Jianwu Li, and Zhaoming Lu. 2021. ST-Tran: Spatial-temporal transformer for cellular traffic prediction. *IEEE Communications Letters* 25, 10 (2021), 3325–3329.
- 1570
- 1571 [62] Zhifeng Liu, Zhiliang Wang, Xia Yin, Xingang Shi, Yingya Guo, and Ying Tian. 2019. Traffic matrix prediction based on deep learning for dynamic traffic engineering. In *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 1–7.
- 1572
- 1573 [63] Iraj Lohrasbinasab, Amin Shahraki, Amir Taherkordi, and Anca Delia Jurcut. 2021. From statistical-to machine learning-based network traffic prediction. *Transactions on Emerging Telecommunications Technologies* (2021), e4394.
- 1574
- 1575 [64] Álvaro López-Raventós, Francesc Wilhelmi, Sergio Barrachina-Muñoz, and Boris Bellalta. 2018. Machine learning and software defined networks for high-density wlangs.
- 1576
- 1577 [65] Rishabh Madan and Partha Sarathi Mangipudi. 2018. Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*. IEEE, 1–5.
- 1578
- 1579 [66] Attila M Nagy and Vilmos Simon. 2018. Survey on traffic prediction in smart cities. *Pervasive and Mobile Computing* 50 (2018), 148–163.
- 1580
- 1581 [67] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, Dusit Niyato, Octavia Dobre, and H Vincent Poor. 2021. 6G Internet of Things: A comprehensive survey. *IEEE Internet of Things Journal* (2021).
- 1582
- 1583 [68] Laisen Nie, Xiaojie Wang, Shupeng Wang, Zhaolong Ning, Mohammad S Obaidat, Balqies Sadoun, and Shengtao Li. 2021. Network traffic prediction in industrial Internet of Things backbone networks: A multitask learning mechanism. *IEEE Transactions on Industrial Informatics* 17, 10 (2021), 7123–7132.
- 1584
- 1585 [69] Tiago Prado Oliveira, Jamil Salem Barbar, and Alessandro Santos Soares. 2014. Multilayer perceptron and stacked autoencoder for Internet traffic prediction. In *IFIP International Conference on Network and Parallel Computing*. Springer, 61–71.
- 1586
- 1587 [70] Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. 2018. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Communications Surveys & Tutorials* 21, 2 (2018), 1988–2014.
- 1588
- 1589 [71] Sonali Appasaheb Patil, L Arun Raj, and Bhupesh Kumar Singh. 2021. Prediction of IoT traffic using the gated recurrent unit neural network-(GRU-NN-) based predictive model. *Security and Communication Networks* 2021 (2021), 1–7.
- 1590
- 1591
- 1592 [72] Vasileios Perifanis, Nikolaos Pavlidis, Remous-Aris Koutsiamanis, and Pavlos S Efraimidis. 2022. Federated Learning for 5G Base Station Traffic Forecasting. *arXiv preprint arXiv:2211.15220* (2022).
- 1593
- 1594 [73] Vasileios Perifanis, Nikolaos Pavlidis, Selim F Yilmaz, Francesc Wilhelmi, Elia Guerra, Marco Miozzo, Pavlos S Efraimidis, Paolo Dini, and Remous-Aris Koutsiamanis. 2023. Towards Energy-Aware Federated Traffic Prediction for Cellular Networks. In *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 93–100.
- 1595
- 1596 [74] Quoc-Viet Pham, Kapal Dev, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, Thien Huynh-The, et al. 2021. Fusion of federated learning and industrial Internet of Things: A survey. *arXiv preprint arXiv:2101.00798* (2021).
- 1597
- 1598 [75] Hnin Pann Phyu, Diala Naboulsi, and Razvan Stanica. 2022. Mobile traffic forecasting for network slices: A federated-learning approach. In *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 745–751.
- 1599
- 1600
- 1601 [76] Hnin Pann Phyu, Razvan Stanica, and Diala Naboulsi. 2023. Multi-slice privacy-aware traffic forecasting at RAN level: A scalable federated-learning approach. *IEEE Transactions on Network and Service Management* (2023).
- 1602
- 1603 [77] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and SS Iyengar. 2018. "A survey on deep learning: Algorithms, techniques, and applications". *ACM Computing Surveys (CSUR)* 51, 5 (2018), 1–36.
- 1604
- 1605 [78] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. 2020. Binary neural networks: A survey. *Pattern Recognition* 105 (2020), 107281.
- 1606
- 1607 [79] Youyang Qu, Xin Yuan, Ming Ding, Wei Ni, Thierry Rakotoarivelo, and David Smith. 2023. Learn to Unlearn: A Survey on Machine Unlearning. *arXiv preprint arXiv:2305.07512* (2023).
- 1608
- 1609 [80] Arcangela Rago, Giuseppe Piro, Gennaro Boggia, and Paolo Dini. 2020. Multi-Task Learning at the Mobile Edge: An Effective Way to Combine Traffic Classification and Prediction. *IEEE Transactions on Vehicular Technology* 69, 9 (2020), 10362–10374.
- 1610
- 1611 [81] Nipun Ramakrishnan and Tarun Soni. 2018. Network traffic prediction using recurrent neural networks. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 187–193.
- 1612
- 1613 [82] Shahbaz Rezaei and Xin Liu. 2020. Multitask learning for network traffic classification. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–9.
- 1614
- 1615 [83] Jorai Rijdsdijk, Lichao Wu, Guilherme Perin, and Stjepan Picek. 2021. Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2021), 677–707.
- 1616
- 1617

- 1618 [84] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz-Rodríguez. 2021.
 1619 Explainable artificial intelligence (xai) on timeseries data: A survey. *arXiv preprint arXiv:2104.00950* (2021).
- 1620 [85] Alessio Sacco, Flavio Esposito, and Guido Marchetto. 2020. A Federated Learning Approach to Routing in Challenged
 1621 SDN-Enabled Edge Networks. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 150–154.
- 1622 [86] Sajal Saha, Anwar Haque, and Greg Sidebottom. 2022. Transfer Learning Based Efficient Traffic Prediction with
 1623 Limited Training Data. *arXiv preprint arXiv:2205.04344* (2022).
- 1624 [87] Wenxin Shen, Haixia Zhang, Shuaishuai Guo, and Chuanting Zhang. 2021. Time-Wise Attention Aided Convolutional
 1625 Neural Network for Data-Driven Cellular Traffic Prediction. *IEEE Wireless Communications Letters* (2021).
- 1626 [88] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2019. The performance of LSTM and BiLSTM in
 1627 forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 3285–3292.
- 1628 [89] Faranaksadat Solat, Tae Yeon Kim, and Joohyung Lee. 2023. A novel group management scheme of clustered federated
 1629 learning for mobile traffic prediction in mobile edge computing systems. *Journal of Communications and Networks*
 1630 (2023).
- 1631 [90] Tejas Subramanya and Roberto Riggio. 2021. Centralized and federated learning for predictive VNF autoscaling in
 1632 multi-domain 5G networks and beyond. *IEEE Transactions on Network and Service Management* (2021).
- 1633 [91] Nasrin Sultana, Naveen Chilamkurti, Wei Peng, and Rabei Alhadad. 2019. Survey on SDN based network intrusion
 1634 detection system using machine learning approaches. *Peer-to-Peer Networking and Applications* 12, 2 (2019), 493–501.
- 1635 [92] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv*
 1636 *preprint arXiv:1409.3215* (2014).
- 1637 [93] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep
 1638 transfer learning. In *International conference on artificial neural networks*. Springer, 270–279.
- 1639 [94] Hoang Duy Trinh, Lorenza Giupponi, and Paolo Dini. 2018. Mobile traffic prediction from raw data using LSTM
 1640 networks. In *2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications*
 1641 *(PIMRC)*. IEEE, 1827–1832.
- 1642 [95] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A
 1643 hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial*
 1644 *Intelligence and Security*. 1–11.
- 1645 [96] Theodoros Tsiolakis, Nikolaos Pavlidis, Vasileios Perifanis, and Pavlos S Efraimidis. 2024. Carbon-Aware Machine
 1646 Learning: A Case Study on Cellular Traffic Forecasting with Spiking Neural Networks. In *IFIP International Conference*
 1647 *on Artificial Intelligence Applications and Innovations*. Springer, 178–191.
- 1648 [97] Paul Tune, Matthew Roughan, H Haddadi, and O Bonaventure. 2013. Internet traffic matrices: A primer. *Recent*
 1649 *Advances in Networking* 1 (2013), 1–56.
- 1650 [98] Steve Uhlig, Bruno Quoitin, Jean Lepropre, and Simon Balon. 2006. Providing public intradomain traffic matrices to
 1651 the research community. *ACM SIGCOMM Computer Communication Review* 36, 1 (2006), 83–86.
- 1652 [99] Muhammad Usama, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala
 1653 Al-Fuqaha. 2019. Unsupervised machine learning for networking: Techniques, applications and research challenges.
 1654 *IEEE Access* 7 (2019), 65579–65615.
- 1655 [100] Krishna Viswanatham Veerubhotla et al. 2022. Origin Destination Traffic Matrix Prediction in Networks using
 1656 Recurrent Layer Algorithms. In *2022 IEEE 6th Conference on Information and Communication Technology (CICT)*. IEEE,
 1657 1–5.
- 1658 [101] Yeshwanth Venkatesha, Youngeun Kim, Leandros Tassioulas, and Priyadarshini Panda. 2021. Federated learning with
 1659 spiking neural networks. *IEEE Transactions on Signal Processing* 69 (2021), 6183–6194.
- 1660 [102] Jing Wang, Jian Tang, Zhiyuan Xu, Yanzhi Wang, Guoliang Xue, Xing Zhang, and Dejun Yang. 2017. Spatiotemporal
 1661 modeling and prediction in cellular networks: A big data enabled deep learning approach. In *IEEE INFOCOM 2017-IEEE*
 1662 *Conference on Computer Communications*. IEEE, 1–9.
- 1663 [103] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang. 2017. Machine learning for networking: Workflow,
 1664 advances and opportunities. *IEEE Network* 32, 2 (2017), 92–99.
- 1665 [104] Xu Wang, Zimu Zhou, Fu Xiao, Kai Xing, Zheng Yang, Yunhao Liu, and Chunyi Peng. 2018. Spatio-temporal analysis
 1666 and prediction of cellular traffic in metropolis. *IEEE Transactions on Mobile Computing* 18, 9 (2018), 2190–2202.
- [105] Zi Wang, Jia Hu, Geyong Min, Zhiwei Zhao, Zheng Chang, and Zhe Wang. 2022. Spatial-Temporal Cellular Traffic
 Prediction for 5 G and Beyond: A Graph Neural Networks-Based Approach. *IEEE Transactions on Industrial Informatics*
 (2022).
- [106] Zihuan Wang and Vincent WS Wong. 2022. Cellular Traffic Prediction Using Deep Convolutional Neural Network
 with Attention Mechanism. In *ICC 2022-IEEE International Conference on Communications*. IEEE, 2339–2344.
- [107] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time series data
 augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478* (2020).

- 1667 [108] Qiong Wu, Kaiwen He, Xu Chen, Shuai Yu, and Junshan Zhang. 2021. Deep transfer learning across cities for mobile
1668 traffic prediction. *IEEE/ACM Transactions on Networking* 30, 3 (2021), 1255–1267.
- 1669 [109] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive
1670 survey on graph neural networks. *IEEE transactions on neural networks and learning systems* (2020).
- 1671 [110] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-
1672 temporal graph modeling. *arXiv preprint arXiv:1906.00121* (2019).
- 1673 [111] Junfeng Xie, F Richard Yu, Tao Huang, Renchao Xie, Jiang Liu, Chenmeng Wang, and Yunjie Liu. 2018. A survey of
1674 machine learning techniques applied to software defined networking (SDN): Research issues and challenges. *IEEE*
1675 *Communications Surveys & Tutorials* 21, 1 (2018), 393–430.
- 1676 [112] Liyan Xiong, Xiangzheng Ling, Xiaohui Huang, Hong Tang, Weimin Yuan, and Weichun Huang. 2020. A sparse
1677 connected long short-term memory with sharing weight for time series prediction. *IEEE Access* 8 (2020), 66856–66866.
- 1678 [113] Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. 2021. A survey on green deep learning. *arXiv*
1679 *preprint arXiv:2111.05193* (2021).
- 1680 [114] Li Yang, Xiangxiang Gu, and Huaifeng Shi. 2020. A Noval Satellite Network Traffic Prediction Method Based on
1681 GCN-GRU. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 718–723.
- 1682 [115] Jiexia Ye, Juanjuan Zhao, Kejiang Ye, and Chengzhong Xu. 2020. How to build a graph-based deep learning architecture
1683 in traffic domain: A survey. *IEEE Transactions on Intelligent Transportation Systems* (2020).
- 1684 [116] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. 2020. A comprehensive survey on traffic
1685 prediction. *arXiv preprint arXiv:2004.08555* (2020).
- 1686 [117] Lixing Yu, Ming Li, Wenqiang Jin, Yifan Guo, Qianlong Wang, Feng Yan, and Pan Li. 2020. Step: A spatio-temporal
1687 fine-granular user traffic prediction system for cellular networks. *IEEE Transactions on Mobile Computing* (2020).
- 1688 [118] Shah Zeb, Muhammad Ahmad Rathore, Aamir Mahmood, Syed Ali Hassan, JongWon Kim, and Mikael Gidlund.
1689 2021. Edge Intelligence in Softwarized 6G: Deep Learning-enabled Network Traffic Predictions. *arXiv preprint*
1690 *arXiv:2108.00332* (2021).
- 1691 [119] Qingtian Zeng, Qiang Sun, Geng Chen, Hua Duan, Chao Li, and Ge Song. 2020. Traffic Prediction of Wireless Cellular
1692 Networks Based on Deep Transfer Learning and Cross-Domain Data. *IEEE Access* 8 (2020), 172387–172397.
- 1693 [120] Chuanting Zhang, Shuping Dang, Basem Shihada, and Mohamed-Slim Alouini. 2021. Dual Attention-Based Federated
1694 Learning for Wireless Traffic Prediction. *IEEE INFOCOM 2017-IEEE Conference on Computer Communications* (2021).
- 1695 [121] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. 2019. Deep learning in mobile and wireless networking: A survey.
1696 *IEEE Communications surveys & tutorials* 21, 3 (2019), 2224–2287.
- 1697 [122] Chuanting Zhang, Haixia Zhang, Jingping Qiao, Dongfeng Yuan, and Minggao Zhang. 2019. Deep transfer learning for
1698 intelligent cellular traffic prediction based on cross-domain big data. *IEEE Journal on Selected Areas in Communications*
1699 37, 6 (2019), 1389–1401.
- 1700 [123] Chuanting Zhang, Haixia Zhang, Dongfeng Yuan, and Minggao Zhang. 2018. Citywide cellular traffic prediction
1701 based on densely connected convolutional neural networks. *IEEE Communications Letters* 22, 8 (2018), 1656–1659.
- 1702 [124] Jianwei Zhang, Xinhua Hu, Zengyu Cai, Liang Zhu, and Yuan Feng. 2023. Federated Learning Based on Mutual
1703 Information Clustering for Wireless Traffic Prediction. *Electronics* 12, 21 (2023), 4476.
- 1704 [125] Liang Zhang, Chuanting Zhang, and Basem Shihada. 2022. Efficient wireless traffic prediction at the edge: A federated
1705 meta-learning approach. *IEEE Communications Letters* 26, 7 (2022), 1573–1577.
- 1706 [126] Qingchen Zhang, Laurence T Yang, Zhikui Chen, and Peng Li. 2018. "A survey on deep learning for big data".
1707 *Information Fusion* 42 (2018), 146–157.
- 1708 [127] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).
- 1709 [128] Zhenwei Zhang, Leon Yan, and Yuantao Gu. 2023. ST2T: A Spatio-Temporal Transformer for Cellular Traffic Prediction
1710 in Digital Twin Systems. In *2023 IEEE 6th International Conference on Electronic Information and Communication*
1711 *Technology (ICEICT)*. IEEE, 1112–1117.
- 1712 [129] Shuai Zhao, Xiaopeng Jiang, Guy Jacobson, Rittwik Jana, Wen-Ling Hsu, Raif Rustamov, Manoop Talasila, Syed Anwar
1713 Aftab, Yi Chen, and Cristian Borcea. 2020. Cellular Network Traffic Prediction Incorporating Handover: A Graph
1714 Convolutional Approach. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking*
1715 *(SECON)*. IEEE, 1–9.
- 1716 [130] Weiping Zheng, Yiyong Li, Minli Hong, Xiaomao Fan, and Gansen Zhao. 2022. Flow-by-flow traffic matrix prediction
1717 methods: Achieving accurate, adaptable, low cost results. *Computer Communications* 194 (2022), 348–360.
- 1718 [131] Jian Zhou, Taotao Han, Fu Xiao, Guan Gui, Bamidele Adebisi, Haris Gacananin, and Hikmet Sari. 2022. Multiscale
1719 network traffic prediction method based on deep echo-state network for internet of things. *IEEE Internet of Things*
1720 *Journal* 9, 21 (2022), 21862–21874.
- 1721 [132] Jian Zhou, Haoming Wang, Fu Xiao, Xiaoyong Yan, and Lijuan Sun. 2021. Network traffic prediction method based
1722 on echo state network with adaptive reservoir. *Software: Practice and Experience* 51, 11 (2021), 2238–2251.