Received: 22 July 2021 | Accepted: 6 September 2021

DOI: 10.1002/ett.4394

SURVEY PAPER

WILEY

From statistical- to machine learning-based network traffic prediction

Iraj Lohrasbinasab[1] | Amin Shahraki[2,3] | Amir Taherkordi[2] | Anca Delia Jurcut[3]

[1]Faculty of Computer Sciences, Østfold University College, Halden, Norway

[2]Department of Informatics, University of Oslo, Oslo, Norway

[3]School of Computer Science, University College Dublin, Dublin, Ireland

Correspondence
Amin Shahraki, School of Computer Science, University College Dublin, Dublin, Ireland.
Email:am.shahraki@ieee.org

Abstract

Nowadays, due to the exponential and continuous expansion of new paradigms such as Internet of Things (IoT), Internet of Vehicles (IoV) and 6G, the world is witnessing a tremendous and sharp increase of network traffic. In such large-scale, heterogeneous, and complex networks, the volume of transferred data, as *big data*, is considered a challenge causing different networking inefficiencies. To overcome these challenges, various techniques are introduced to monitor the performance of networks, called Network Traffic Monitoring and Analysis (NTMA). Network Traffic Prediction (NTP) is a significant subfield of NTMA which is mainly focused on predicting the future of network load and its behavior. NTP techniques can generally be realized in two ways, that is, statistical- and Machine Learning (ML)-based. In this paper, we provide a study on existing NTP techniques through reviewing, investigating, and classifying the recent relevant works conducted in this field. Additionally, we discuss the challenges and future directions of NTP showing that how ML and statistical techniques can be used to solve challenges of NTP.

1 | INTRODUCTION

During the last decades, new networking paradigms, for example, Wireless Sensor Networks (WSNs), Internet of Things (IoT), Internet of Vehicles (IoV) and 6th generation of cellular networks (6G)[1] have been emerging to establish the network infrastructures for real-world applications, for example, smart cities, crisis management, smart roads, etc..[2] Thanks to miniaturization of digital equipment, today's networks include thousands of connected User Equipments (UEs) (known as end node devices) that can generate and/or consume data. IoT as a new emerging networking paradigm, provides an overlay network on top of other network infrastructures, from Near-Field Communication (NFC) to cellular networks to connect a virtually unlimited number of UEs.[3] It is expected that in 2025, the number of connected IoT devices will increase to 75 billion as predicted by Cisco.[4,5] While managing such numerous devices is a challenging issue, other characteristics of networks, for example, heterogeneity and mobility, can cause inefficiency in networking. Network heterogeneity is not solely due to the diversity of device types, as it can also be related to some other factors such as the volume of data generated by each connected UEs, the required services, and the diversity of network connections. Given these characteristics of new networking paradigms, the volume of data generated may be very huge which has given rise to the Big Data era.[6]

To provide an efficient network infrastructure to transfer and manage such a huge volume of data, different techniques are introduced, mainly to prevent various network faults and inefficiencies, support Quality of Service (QoS) and provide security. Apart from the type of the network infrastructure, the QoS depends on the performance of route of data, and even the ability of routing devices to analyze real-time network situation to make dynamic networking adjustments and allocation decisions. Network Traffic Monitoring and Analyzing (NTMA) techniques are mainly introduced to monitor the performance of networking by providing information to analyze the network and offer solutions to address the challenges without human intervention.[7] There are four main subfields in the NTMA including[8] (i) Network Traffic Prediction (NTP),[9] (ii) Network Traffic Classification (NTC),[10] (iii) Fault Management, and (iv) Network Security. Among all these subfields, NTP focuses on analyzing the network load and prediction of the network traffic to avoid faults and inefficiencies in networking. In this study, we focus on NTP, as one of the most critical solutions to address various networking challenges, for example, resource provisioning, congestion control, resource discovery, network behavior analysis, and anomaly detection.[6]

Different techniques are introduced to perform NTP, but generally, existing solutions can be divided into two types, that is, Machine Learning (ML)-based techniques and statistical-based techniques. As NTP can be designed based on both types, we first review the most relevant techniques and then investigate the proposed solutions for each type. The contributions of this study include:

- Investigating the existing NTP techniques and available solutions to predict the network behavior.
- Classifying the NTP techniques based on statistical-, ML-based, and hybrid techniques.
- Providing a concrete future direction based on real-world applications compared to the state of the art techniques, models, and frameworks.
- Proposing a schema to integrate statistical-based techniques and ML-based techniques to improve the performance of NTP techniques.

The rest of the paper is summarized as follows. In Section 2, we first introduce the basic concepts and discuss the available types of techniques for the NTP. In Section 3, we survey and analyze existing solutions and provide a classification of them. In Section 4, we discuss the challenges and future directions, and finally, in Section 5, we conclude this study.

2 | BASIC CONCEPTS

This section briefly explains some key concepts of NTP and existing types of solutions. We first introduce the NTP concept and then continue this section by discussing and comparing the statistical- and ML-based techniques which are further reviewed in Section 3.

2.1 | Network traffic prediction

The continuous and exponential growth in scale, speed, and heterogeneity of network traffic streams, as seen in networking platforms such as IoT and 5G, as well as emerging networking technologies like Software-Defined Network (SDN), Network Function Virtualization (NFV), and Fog computing has proven the desperate need to NTMA solutions.[8,11] On the foundation laid down by Big Data Analytics, and using techniques such as Traffic Prediction, Traffic Classification, Anomaly Detection, and Fault Detection, NTMA is engaged in tackling a range of substantial problems. To name a few, many aspects of network management including fault management, configuration management, accounting management, security management, and performance management achieve effectiveness by utilizing the NTMA capabilities.[12]

At the core of the mentioned techniques, sensing the network status in a real-time manner and analyzing the trend of changes based on the data extracted from the network traffic plays a key role. Maintaining user satisfaction at the desired level (known as Quality of Experience [QoE]) is one of the main goals of performance management which can be achieved by continuously monitoring the status of the networking functions. Considering the inherent relationships between user-oriented QoE and network-oriented QoS parameters, the performance management objectives could be achieved through monitoring and measuring QoS parameters such as throughput, delay, jitter, or loss rate.[13]

The network traffic data can be extracted from the packet inspection process by one of the Deep Packet Inspection (DPI) or Shallow Packet Inspection (SPI) methods.[12] While the first is based on reading and analyzing the full packet contents that include application headers and payload, the second method examines only headers of the network and/or transport layers in those packets selected based on the adopted sampling strategies.[7] Nevertheless, apart from some specific applications of DPI, in cases such as filtering or troubleshooting—given the lack of feasibility of applying this method due to privacy challenges and imposing significant computational and memory overheads on the network—the real-time Traffic-Monitoring-Centric tasks would rely on SPI for extracting the required data from packet streams.[12,14]

NTP as a subtechnique of NTMA is used to determine the status of the network, identify changes and predict the network traffic behavior in a foreseeable future. Generally, the results of NTP techniques can be used in a wide range of applications, for example, QoS provisioning, fault detection, and security attack detection. The problem of predicting future network traffic volume is traditionally formulated in the form of a Time Series Forecasting (TSF) or rarely a spatiotemporal problem aimed at constructing a regression model capable of estimating future traffic volume by extracting the existing dependencies between historical and future data.[15,16] Typically, low computational overhead, simplicity, and the limited number of features can be referred to as advantages of TSF approaches.[17] On the other hand, due to new demands and exigencies stemmed from the ever-increasing rise in scale, speed, and heterogeneity of networks, non-TSF approaches are becoming more and more propounded.[18] These methods typically leverage flow and packet header data to estimate future incoming flows instead of traffic volume.[19]

Basically, popular NTP solutions are divided into statistical analysis methods and ML-based models.[20,21] In the following, we shed light on some differentiation aspects of common network prediction's methods and techniques.

2.2 | Statistical techniques for NTP

Statistical techniques are mainly based on analyzing the data patterns without having any prior knowledge (without training). Most of such techniques compare the current situation of the data, that is, the pattern of the data with the last identified pattern to recognize important changes. Linear statistical-based models extract patterns from the historical data as well as predict future points in time-series according to the lagged data. The well-known members of this category are Autoregression Moving Average (ARMA) and AutoRegressive Integrated Moving Average (ARIMA), as well as variants of the latter.[22] ARIMA, also known as Box-Jenkins model, is a prevalent paradigm among statistical models for time-series prediction. Both of ARMA and ARIMA models are emerged from the convergence of autoregressive (AR) model which involves with lagged values of observations, and moving average (MA) model which takes lagged errors. Nevertheless, The distinction between them is in their approach to the notion of stationarity in time-series. While ARMA assumes the time-series is stationary, ARIMA would provide the stationarity of data through differencing process which might be applied multiple times, until establishment of stationarity. In "ARIMA" the I (stands for "Integrated") refers to this procedure. ARIMA model is denoted as ARIMA(p, d, q), where p indicates the order of the AR part, d shows the involved differencing degree, and q is the order of the moving average part.[23,24] As concisely described below, ARMA and ARIMA can be formulated, respectively, in Equations (2) and (3). Assuming a time series data X_t consisting of real numbers (x_t) and an integer index (t), an ARMA (p,q) model is given by Equation (1) where α_i are the parameters of the (AR) model, θ_i represents the (MA) model's parameters, and ε_t are error terms assumed to be independent and identically distributed (i,i,d) variables sampled from a normal distribution with zero mean.

$$X_t - \alpha_1 X_{t-1} - \cdots - \alpha_p X_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}. \tag{1}$$

Interchangeably, it can be shown as below where L is the lag operator:

$$\left(1 - \sum_{i=1}^{p} \alpha_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t. \tag{2}$$

Interested readers are referred to Reference 23. Assuming the ARIMA model as the generalization of ARMA, the following formula defines an ARIMA(p,d,q) process given drift $\left(\frac{\delta}{1-\sum \varphi_i}\right)$:

$$\left(1 - \sum_{i=1}^{p} \varphi_i L^i\right) (1-L)^d X_t = \delta + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t. \tag{3}$$

Diverse variations of ARIMA are proposed according to various applications and time-series; Among them, it is worth mentioning "Seasonal-ARIMA (SARIMA)"[25] and *"Fractional AutoRegressive Integrated Moving Average (FARIMA)"*.[26] The former is often used in NTP considering its compatibility with the nature of changes in networks which usually obeys certain time patterns. A FARIMA forecasting model is an extension of the ARIMA (p, d, q) model in which the fractional parameter d can take real values rather than just integers, and is given by the equation:

$$\Phi_p(L)(1-L)^d\,(Y_t) = \Theta_q(L)\varepsilon_t, \tag{4}$$

where L is the lag operator, $\Phi_p(L) = 1 - \varphi_1 L - \cdots - \varphi_p L^p$ and $\Theta_q(L) = 1 + \theta_1 L + \cdots + \theta_q L^q$, $(1-L)^d = \sum_{j=0}^{\infty} \begin{pmatrix} d \\ j \end{pmatrix} (-1)^j L^j$ with $\begin{pmatrix} d \\ j \end{pmatrix}(-1) = \frac{\Gamma(-d+j)}{\Gamma(-d)\Gamma(j+1)}$ and $\varepsilon_t \sim\ N\left(0, \sigma^2\right)$ as the error terms.[27]

In addition to the extant components of the conventional ARIMA, the SARIMA[28] model also includes the frequency component (known as *seasonality* and shown by (S).[29]

An SARIMA model would conduct prediction based on a linear combination of past observations and their related errors. As its name implies, the seasonality factor plays a key role in the structure and performance of this model. The SARIMA process often would be shown as models in the form of $SARIMA(p, d, q) \times (P, D, Q)S$. For given time series $\{X_t\}$ with seasonality length (S), the SARIMA process is indicated by the Equation (5), while the differenced series $w_t = (1-B)^d(1-B^S)^D X_t$ is a stationary ARMA process and, d and D are nonnegative integer values.

$$\phi_p(B)\Phi_P(B^S)w_t = \theta_q(B)\Theta(B^S)\epsilon_t, t = 1, 2, \ldots, n, \tag{5}$$

where:

- n is the number of observations up to time t and the backshift operator B is defined as: $B^a W_t = W_{t-a}$.
- ϵ_t is defined as i.i.d samples with a zero mean and variance σ^2, which for all $K \neq 0$ we have: $\mathrm{Cov}(\epsilon_t, \epsilon_{t-k}) = 0$
- The nonseasonal components are: $\begin{cases} \bullet \text{ AR } : \phi(B) = 1 - \phi_1 B - \ \ldots\ - \phi_p B^p \\ \bullet \text{ MA } : \theta(B) = 1 - \theta_1 B - \ \ldots\ - \theta_q B^q \end{cases}$
- The seasonal components are: $\begin{cases} \bullet \text{ SeasonalAR } : \Phi(B^S) = 1 - \Phi_1 B^S - \ \ldots\ - \Phi_P B^{PS} \\ \bullet \text{ SeasonalMA } : \Theta(B^S) = 1 - \Theta_1 B^S - \ \ldots\ - \Theta_Q B^{QS} \end{cases}$

As an another ARIMA's extensions, FARIMA is a generalization of the ARMA model customized to support those applications like NTP in which, besides short-term dependencies, there are considerable linear long-term dependencies between the observations. Unlike the ordinary ARIMA process, the difference parameter (d) in FARIMA model could take noninteger values.[30] The general FARIMA process is expressed in Equation (6) where B is the backshift operator.

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right)(1-B)^d X_t = \left(1 + \sum_{i=1}^{q} \theta_i B^i\right)\varepsilon_t. \tag{6}$$

Generally, the family of ARIMA models is drawn on the assumption of time-series data stationarity; while, in dynamic environments such as IoT, network traffic with severe and intermittent fluctuations could lead to degrading model performance. Nonetheless, by using some transformations other than differentiation (eg, logarithms) to decrease nonstationarity in input data, this deficiency can be overcome to some extent.[23,31]

One of the other standard statistical models widely employed in time-series problems is that of Generalized Autoregressive Conditional Heteroskedasticity (GARCH).[32] This model is an extension of the Autoregressive Conditional Heteroskedasticity (ARCH) model innovated by Engle in 1982 to estimate the target variables' volatility.[33] The main goal is to model the changes in variance of target variables whose part of the total variance is conditioned on lagged values of target variance and model's residuals. To this end, the concept of Conditional Variance (also referred to as Conditional Volatility) plays a key role. Considering $\{\epsilon_t\}$ as a real-valued discrete-time stochastic process, as:

$$\epsilon_t = \sigma_t w_t,$$

where w_t is discrete white noise given i.i.d ($\mu = 0, \sigma^2 = 1$), the GARCH(p,q) process is then denoted by Equation (7).

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{7}$$

where α_i and β_j are model's parameters meanwhile, to avoid negative variance the following constraints are imposed:[34,35]

$$\left\{p \geq 0, q > 0, \alpha_0 > 0, \alpha_i \geq 0, i = 1, \dots, q, \beta_j \geq 0, j = 1, \dots, p\right\}.$$

Moreover, parallel to applying jointly with some nonlinear approaches (eg, ARIMA), some hybrid models based on the ARIMA's foundation have been proposed. For instance, Fuzzy-AutoRegressive Integrated Moving Average (Fuzzy-ARIMA) is an invented method which fuzzifies ARIMA's parameters using the fuzzy regression method.[36] In terms of acronyms, as seen sporadically in some references, Fuzzy-ARIMA is also referred to as FARIMA; which should not be confused with the Fractional-ARIMA model. Meanwhile, The latter is recorded as ARFIMA in some sources as well.[37]

2.3 | ML techniques for NTP

In general, the logical framework of problems solvable by ML techniques can be formulated in four broad categories, namely, classification and regression, clustering, and rule extraction.[19] There are four ML paradigms in the same vein corresponding to the nature of the problem at hand, namely Supervised Learning (SL), Unsupervised, Semi-supervised, and Reinforcement Learning,[38] respectively. Each of these paradigms has its own different effects on proceeding data collection, ground truth creation, and feature engineering. Most of ML methods used in NTP are subtechniques of SL as the models need to be trained by historical data. SL uses labeled data (historical data) to build up the models employed in classification and regression problems, where predicting the outcomes in the form of discrete or continuous quantities is intended. In many real-world problems, access to labeled data is subject to constraints. In networking, most of data gathered from a network is unlabeled or semi-labeled.[39] In the lack of sufficient knowledge or the abundance of missing labels, the Semi-Supervised Learning (SSL) paradigm as a particular variant of SL exploiting different techniques such as Active Learning[40] can be leveraged.[6]

One of the critical aspects in ML is choosing the proper model from the mass of available algorithms and techniques. Different factors can be applied, for example, goals of applications, pros and cons of the operating environment regarding the deployments and applications of ML models, the learning method (ie, supervised or unsupervised), how to access data, etc.[41] Some of the most widely used models are presented in the following:

- *Neural Networks:* Artificial Neural networks (ANNs) are among the most potent and widely used ML techniques.[42] Thanks to the Activate Function, ANNs can learn complex nonlinear dependencies among numerous variables, thus they are generally known as Universal Function Approximators.[42] The general architecture of ANNs is a directed graph[43] consisting of input and output layers, which are connected via the so-called hidden layer, which itself could be consist of one or more layers. Input values reach the output layer by applying transformations through the hidden layers. The number of these layers is also referred to as the *depth* of the model. Based on the ANNs' depth, the idiom "Deep Neural Network" denotes ANNs constructed of two or more hidden layers, opposed to "Shallow Neural Network" referring to the traditional baseline ANNs.[42] Due to their flexible structure, Deep Neural Networks (DNN) have gained striking popularity in time-series prediction. In this context, Recurrent Neural Network (RNN), by allowing inputs to be recycled in hidden layers around the recurrent connections has been a leaping advance. Different RNN-based architectures can be defined as per adopted activation function and how the neurons connect to each other, namely that of Fully Recurrent Neural Network (FRNN), Bidirectional Neural Networks (BNN), stochastic neural networks, and the well-known paradigm Long short-term memory (LSTM).[44] LSTM as an extension of RNN is proposed to resolve the vulnerability of normal RNNs against the gradient exploding/vanishing problem caused by long-term dependencies.[45-47] LSTM with some innovations in its architecture including a triple gate mechanism to control inputs to cells, and a feedback loop for data retention, can learn long-term dependencies and remove invalid inputs that cause perturbation in cell's outputs.[6,48] In practice, an implemented LSTM model usually consists of a set of blocks where each block contains several LSTM cells.

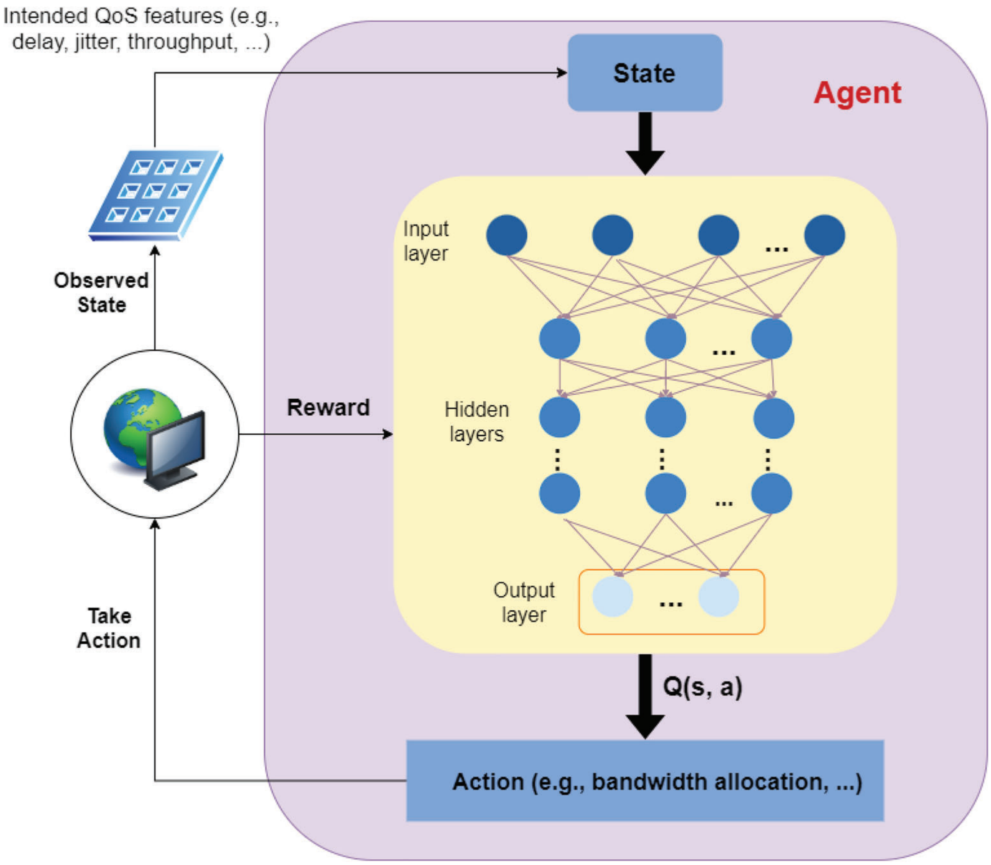
Intended QoS features (e.g.,
delay, jitter, throughput, ...)
State
Agent
Input layer
Hidden layers
Output layer
Observed State
Reward
Take Action
Q(s, a)
Action (e.g., bandwidth allocation, ...)

FIGURE 1 A schema that how deep reinforcement learning interacts with the network

Despite the unavoidable capabilities of deep learning-based methods, the slow training process of these models is a significant problem in their application in dynamic environments. Moreover, the lack of transparency in the learning process of these models is another limitation[49].[50]

- *Reinforcement Learning:* The Q-Learning algorithm, along with Deep Reinforcement Learning (DRL) (which is actually a combination of Q-Learning and DNN), is one of two algorithms representing the Reinforcement Learning (RL) method.[8,12] As elaborated by Watkins,[51] the Q-Learning algorithm provides learner agents that can act optimally in Markovian environments relying on their knowledge stemmed from experiencing actions' consequences, without the need to mapping the environment.[51] It relies on a function called Q-function to learn the table containing all available state-action pairs and their long-term rewards.[52] In NTP, RL and its variants have a good potential to interact with the network to learn the network behavior and predict the behavior of the network in the future. As mentioned, RL can be integrated with DNN which can help to improve the performance of RL techniques. Figure 1 shows a schema of interacting DRL and the network regarding NTP.

Although the number of ML techniques used in NTP is not limited to the list above, DNN and RL are the most important ML techniques based on our literature review explained in Section 3.

2.3.1 | Data collection

Creating an efficient model for a problem is highly dependent on the availability of appropriate and unbiased representative data.[19] Due to the variety of data in different applications of networking as well as alternation of data over time, it is essential to adopt a suitable method for data collection for ML methods to train their models.

In networking, traffic data can be extracted through the packet inspection process by the DPI and SPI methods.[12] While the former is based on reading and, if necessary, analyzing the full packet contents that include application headers and payload, the latter method examines only headers of the network and/or transport layers in those packets selected based on the target sampling strategies.[7] Nevertheless, apart from some specific applications of DPI (eg, filtering or troubleshooting), since this method imposes significant computational and memory overheads on the network, the real-time traffic-monitoring-oriented tasks would rely on SPI for extracting the required data from packet streams.[12,14]

Data collection is performed typically in offline or online manners. In the offline method, data are entirely used for training the model at once, and then, the model is deployed and used for operational data analysis. In the online method, throughout a continuous process, model training would launch in conjunction with deployment in the operating environment, and model knowledge is updated with new input data, which is received in a sequential order.[13,53] As SL, Unsupervised Learning (UL) and SSL are generally used in the offline learning setting, some of RL techniques, for example, State-action-reward-state-action (SARSA)[54] and incremental learning techniques[55] are mainly designed to update the model gradually based on new data (2). Training the model using collected data can be achieved in one of the batch or incremental (also known as streaming) ways depending on the situation and learning settings. In a batch setting, the collected data are divided into three subsets, including training, test, and validation (which in some cases is also called development set). The validation set is used when selecting the appropriate model, and its architecture is part of the process. Otherwise, it would not be needed. Determining the optimal values for the model parameters (eg, the weight of connections between neurons in a neural network [NN]) and evaluating the model's performance would be accomplished, respectively, using the training and test sets. On the other hand, in the incremental method, the data is streamed to the training model for various reasons (eg, large volume and infeasibility of loading at once, or gradual generation of data).[13,53] Moreover, in dynamic environments, especially online applications, the ML model must be continuously retrained. In such cases, to solve the concept drift problem, given the high computational cost of doing it from scratch, using only the new data for training is an efficient solution by incremental approaches.[12] Figure 2 shows different types of collecting data to train ML models in networking.

2.3.2 | Feature engineering

As one of the ML's pillars, feature engineering includes feature selection and feature extraction in addition to data cleansing and data preprocessing. Feature selection refers to selecting the effective discriminator features and removing the irrelevant and redundant features; whereas feature extraction involves extracting new extended features from existing ones. These procedures both would lead to diminishing data dimensionality and computational overhead and, consequently, increasing the model's efficiency and accuracy.[56]

In networking, features can be classified based on the granularity in three levels: packet-level, flow-level, and connection-level.[57] The finest level of granularity is packet-level features where the packet-related statistical data such as mean, root mean square (RMS), variance, as well as time-series information are extracted or derived from collected packets.[19] The independence of these features from the sampling method adopted to collect data is their key advantage.

Features such as mean flow duration, mean of packets per flow, and average flow size in bytes are observable at the flow-level. At the highest level of granularity, there are connection-level features that would be extracted from the transport layer. Throughput and advertised window size in TCP connection headers are examples of these features. Despite the high-quality information provided by connection-level features, the imposition of excessive computational overhead and high distortion in facing with sampling and routing asymmetries are among their drawbacks. Feature extraction is often performed using techniques such as Principal component analysis (PCA), entropy, and Fourier transform.[19] Table 1 shows a summary of some popular network features.

3 | REVIEW OF EXISTING WORK

Using ML techniques and statistical-based techniques in NTP is a well-established research area. Historically, one of the earliest works in applying ML in NTP belongs to Yu and Chen, carried out in 1993. They used multilayer perceptron (MLP) NN (MLP-NN) motivated by enhancing the accuracy over traditional AR methods.[19] Since then, many researchers have lent themselves to improve the ML-based solutions for predicting network behavior as accurately and timely as possible. However, in this survey we focus on the most important recent works in the field. To review the literature, we have used the method shown in Figure 3 to refine the related literature. The NTP-related literature is classified as below.

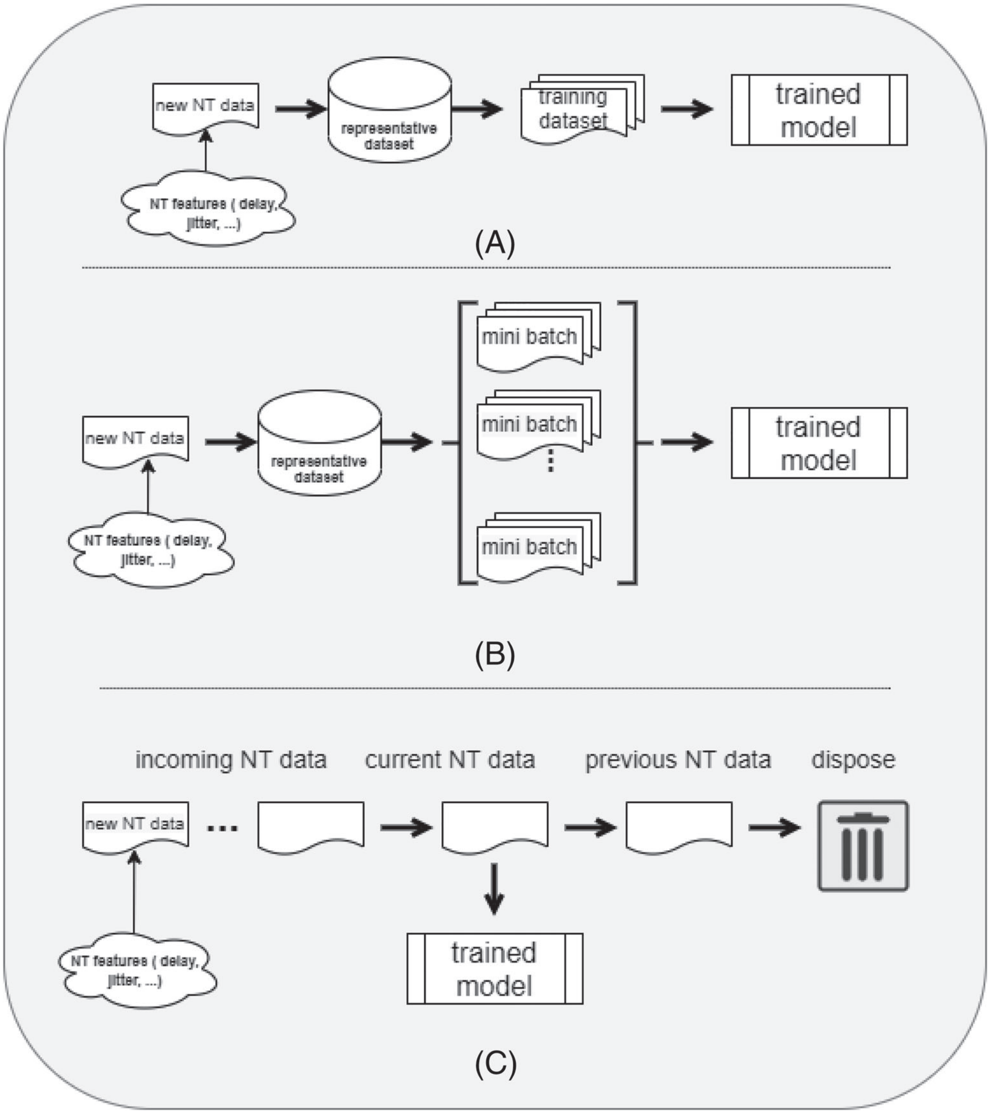

FIGURE 2 Data collection settings. (A) Batch Offline; (B) Batch incremental; (C) Online incremental (streaming)

TABLE 1 More popular features used in machine learning for networking

Feature granularity	Description and examples
Packet-level:	Source IP address
	Destination IP address
	Source host number
	Destination host number
	Frame length
	Frame number
	Packet inter-arrival time
Content-based:	HTTP protocol used to submit data from client to server
Time-based:	Number of frames received by a unique destination in the last T seconds from the same source
	Number of frames received by a unique source in the last *T* seconds from the same destination
Connection-based:	The number of packets flowing from source to destination
	The number of packets flowing from destination to source

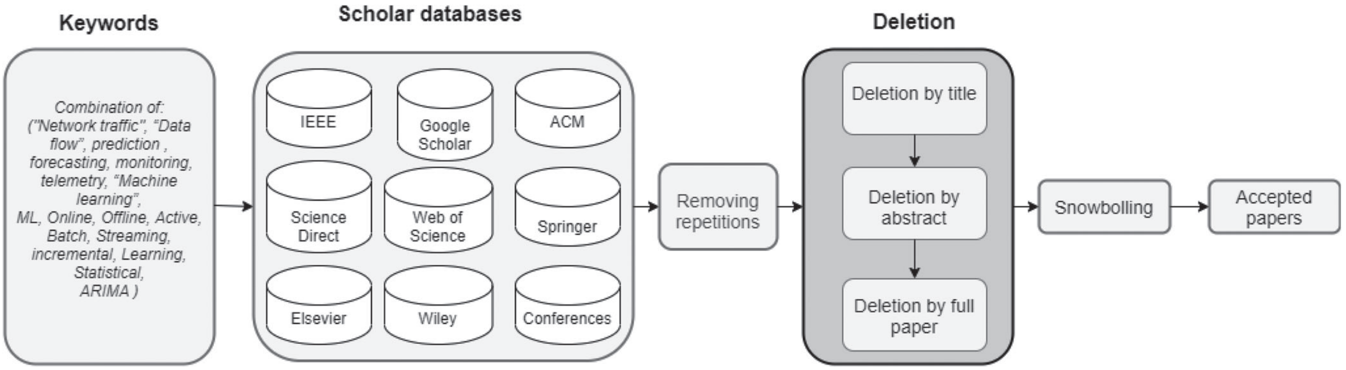

FIGURE 3 The method adopted to refine the literature[58]

3.1 | Pure ML-based NTP solutions

Evaluation of performance and efficiency of ML-based methods is a significant part of the literature. LSTM is one of the most widely used techniques in this field.

The survey by Alawe et al[59] aims at Access and Mobility Management Function (AMF) in the 5G network. To this end, two approaches, Feedforward neural network (FFNN) and LSTM are examined. The authors propose to use ML to forecast the arrival time of requests from User Equipment (UE), and consequently, the scale-out/in process. This avoids the rejection of requests and keeping the attach duration (how long do UE are connected to network resources) low. The results indicate that LSTM outperforms FFNN in terms of prediction accuracy. The dataset is classified into 10 different classes, based on the load and the number of AMFs needed. The first technique used for predicting the class of load of the next period is FFNN. The second technique tested for predicting the average load of the upcoming period of time is LSTM. Both networks are trained with the 60% dataset and then asked to predict the remaining 40%.The results indicate that LSTM outperforms FFNN in terms of prediction accuracy.

Trinh et al[46] have dedicated their work to present a network traffic prediction model in the LTE environment, using the LSTM algorithm. The LSTM network consisting of multiple unified LSTM units is applied to the raw mobile traffic data collected directly from the Physical Downlink Control CHannel (PDCCH) of LTE. By assuming NTP as a supervised multivariate problem, the proposed model aims at minimizing the prediction error with respect to the information extracted from the PDCCH. This data gathering methodology and the multistep structure adopted for the predictive network are emphasized as dedicated research aspects. According to comparison results, the composed model outperforms ARIMA and FFNN models.

Pruning internal connections between NN neurons to diminish computational cost constitutes the idea underlying the research by Hua et al.[47] Based on this, a heuristic architecture with sparse neural connections, called the Random Connectivity Long Short-Term Memory (RCLSTM) is introduced, in which the complete (one-to-one) connection between neural network neurons, as is in conventional LSTM, has given way to a random pattern of links between them. The simulated model consists of a three-layer stack RCLSTM.

Wang et al[60] puts forward a model that aims to improve cellular traffic prediction accuracy with limited real data and support data privacy. The model is called (ctGAN- S2S) and consists of a cellular traffic generative adversarial network and a sequence-to-sequence neural network, in which learning is fed by an arbitrary length of time-window from the historical time-series data of cellular traffic. The augmentation model generates close-to-real cellular traffic data; thus, eliminating the need for original data supports data protection and data privacy.

The main contribution of Vinayakumar et al[20] is to compare some nonlinear prediction methods, namely that of FFNN, RNN, identity recurrent neural network (IRNN), gated recurrent unit (GRU), and LSTM (generally referred to as RNN) in terms of performance in Traffic Matrix (TM) prediction in large networks under different experiments. The arranged experiments in test scenarios help to identify the optimal network parameters and network structure of RNN. The TM is passed by using the sliding window approach. The obtained results indicate the superiority of LSTM over other models; meanwhile, GRU has relatively imposed less computational cost.

In considerable number of researches, Gaussian Process Regression (GPR) is the undergrounding leveraged technique. The conducted research in Bayati et al[61] is one among those who exploit GPR in this context. The proposed solution is based on the Direct (or Parallel) strategy for multiple-step-ahead traffic prediction in which the entire process is divided into H distinct models that are trained concurrently to conduct H-step-ahead time-series predicting. Each time-step forecasting is fed by the gained prediction in the previous time-step, as one of the input features. Subsequently, the prediction error at a time-step (or the uncertainty in the feature vector) is propagated through the forecasts at the next time-step. In order to tackle the error propagation, the paradigm has been investigated, indicating that the desired performance of a multiple-step-ahead prediction is strictly depended on, and could be influenced by the classification of data at higher levels.

Ensuring the QoS in the network in a way that simultaneously provides resource efficiency (bandwidth) requires an accurate real-time estimation of the network's future behavior. To this end, an online bandwidth allocation method based on GPR has been proposed by Kim and Hwang.[62] The theorem that a stationary increase in the size of a given process would lead to large-buffer asymptotics of the queue length process forms the basis of the proposed model for deriving the proper bandwidth.

A combined solution to deal with complex network data flows is presented by Wang et al.[63] The proposed solution consists of a preprocessing step and a 2-fold prediction process using LSTM and GPR models. In the initial phase, the data flow's dominant periodic features are extracted through Fourier analysis, and then the LSTM model is applied to the remaining small components. A complementary step adopting GPR is launched, estimating the residual components to improve the prediction's accuracy.

Poupart et al[64] elaborate utilizing ML techniques to approximate each network flow's size at its start, focused on detecting elephant flows independent of source application or end host. The proposed estimation method benefits the metadata exploited from the first few packets of a flow. In this regard, the authors explain three ML techniques, including Gaussian processes, Gaussian Mixture Model with Bayesian Moment Matching, and neural networks to predict the flow size based on existing historical data and online streaming data.

Mikaeil[65] proposes a method for near-time prediction of primary user (PU) channel state availability (ie, spectrum occupancy) in Cognitive Radio Networks (CRN) using Bayesian online learning (BOL). Given that the nature of the PUs channel state availability can also be considered a dual-states switching time series, the captured time-series representing the Pus channel state (ie, PU idle or PU occupied) are fed as an observations sequence into the BOL prediction algorithm.

The primary use of cardinality estimation algorithms in computer networks is counting the number of distinct flows. Cohen and Nezri[66] have investigated the application of flow cardinality estimation algorithms in SDN environments. their focus was on common deficiencies faced by sampling methods, especially in adapting to changes in flow size distribution. Further, they have introduced and elucidated a framework which benefits from online ML, and to achieve the best performance and accuracy, three popular linear regression ML algorithms, namely that of Stochastic Gradient Descent (SGD), Recursive Least Squares (RLS), and Passive-Aggressive (PA), have been examined.

Cui et al[67] discuss the Stochastic Online Learning (SOL) technique as a tailored model for that of environments like Mobile Edge Computing (MEC), which can experience time-varying, stochastic traffic arrivals without the Markov property. Among the multitude of articles, this paper is one of the few that addresses online learning from network behavior. Unlike the majority of ML methods that rely on learning from the training data, SOL learns from network changes by using SGD. This approach is aimed at establishing a trade-off between learning accuracy and learning time and increasing network throughput.

Zhang et al[68] present an incremental deep computation model for wireless big data feature learning in IoT. The model is constructed by stacking several incremental tensor auto-encoders (ITAE). To handle new arrival wireless samples, two types of ITAEs, based on the learning strategy, are developed, namely the parameter-based incremental learning algorithm (PI-TAE) and the structure-based incremental learning algorithm (SI-TAE). In facing new arrival samples, the proposed model only needs to load the new samples to the memory to update the parameters and structure, respectively, by PI-TAE and SI-TAE. This mechanism underlies this model's capability to deal with wireless big data for feature learning in real-time.

The cellular link bandwidth prediction in Long-Term Evolution (LTE) networks has been investigated by Yue et al.[69] The authors have approached the problem by analyzing the correlations between various lower-layer information and the link bandwidth. Thereupon, they propose a framework using a Random Forest-based prediction model, which through an offline data feeding process, and exploiting the intrinsic capabilities of Random Forest, identifies the most important features, and uses them to predict link bandwidth in real time.

Zhang et al[70] have introduced an approach using Convolutional Neural Network (CNN) to collectively model spatial and temporal dependence for cell traffic prediction. The key features of the proposed approach are considering the traffic data as images, as well as utilizing a parametric matrix-based fusion method to estimate influence degrees of the spatial and temporal dependence.

In a study by Pfulb et al,[71] the problem of estimating the expected bit-rate of network flows based on their metadata has become a three-tier classification problem using a fully connected DNN with the ReLU activation function. Their approach embraces three stages including data collection, data preparation, and data processing. In this approach, DNN training is treated as a streaming problem by dividing the dataset into identical data blocks. The model is trained and tested in a semi-streaming fashion applied on all blocks one by one such that all intended preparations being performed block-wise.

To improve the self-management and active adjustment capabilities of base stations in wireless networks, by Li et al[72] the temporal and spatial correlation of traffic data is addressed. To this end, they have composed a deep network-based framework of network traffic prediction using CNN and LSTM, constituted of three main units by which the spatio-temporal correlation of wireless network traffic data can be captured effectively.

The first part of Table 2 shows the list of the reviewed literature in this category.

3.2 | Pure statistical-based NTP solutions

The following articles mainly deal with the problem using statistical techniques such as ARIMA family and their innovative combinations.

The conducted research by Tran et al[73] is a study on different varieties of exponential smoothing method namely single, double, Holt-Winters No Seasonal, and Holt-Winters Additive Seasonal (that are categorized as simple exponential smoothing methods [S-ESM]), and Error, Trend, Seasonal exponential smoothing method (ETS-ESM) in terms of the effect of their smoothing factors on accuracy of short-term NTP, as well as their suitability for dealing with voice and data over the cellular network. The results show that HWMS outperform other methods.

Rafsanjani et al[22] proposed a method called *QARIMA* as an extension of ARIMA to predict the queues in packet forwarders in networks. Their proposed method determine the last trend of the behavior of the queue and make a linear regression model to predict. They showed that their proposed method can predict the delay of packets with a high accuracy. Shahraki et al[74] integrated and extended the ideas proposed in References 75 and 76 to propose an online network behavior analysis technique for IoT networks. The proposed technique analyses the data gathered from IoT networks to determine and predict important changes that can cause inefficiencies.

Mehdi et al[36] have incorporated fuzzy regression and ARIMA models (Fuzzy-ARIMA) in promising to take both methods' advantages. Besides, to perform real-time predictions based on historical data, they have adopted a sliding window technique called SOFA ,which reduces the effect of input data fluctuations over time.

In promising to allocate bandwidth efficiently in SDNs, Bouzidi et al,[77] introduce a heuristic rules placement algorithm in which an Online-Learning module, designated upon the linear regression, is utilized for the network delay prediction. The overall framework of the proposed solution encompasses the formulation of the flow rules placement as an Integer Linear Program (ILP) targeting to minimize the total network delay, And finally, to solve the defined ILP problem through an excogitated algorithm, that leads to reducing the time complexity and enhancing the estimation accuracy.

The second part of Table 2 shows the list of the reviewed literature in this category.

3.3 | Hybrid NTP solutions

In addition to the pure statistical- and ML- based NTP techniques, there are some proposed models that use both of these approaches simultaneously called Hybrid solutions.

Xu et al[50] have presented an extension of Reference 78, proposing an architecture for traffic prediction in the Cloud radio access network (C-RAN) with the distributed remote radio heads (RRHs) and the centralized Baseband Units BBUs pool bearing lots of parallel Baseband Unit (BBU)s. In such an architecture, an alternating direction method of multipliers (ADMM) and the cross-validation empowered Gaussian Process (GP) framework is performed in which, the parallel BBUs' contribute in training process and the local predictions would be incorporated altogether via cross-validation to create the final prediction. retaining the trade-off between accuracy and time consumption, and scalability are the two focal claims in this method. Bouzidi et al,[79] have composed an extension to Reference 77 with a different experimental

TABLE 2 Comparing existing works on Network Traffic Prediction (NTP)

References	Network	Purpose	Tr.	Approach	Description
Pure ML-based NTP solutions					
59	5G	Access and mobility management	f / b	LSTM and FFNN	Comparison of LSTM and FFNN in predicting network traffic state.
46	LTE	Optimal resource allocation	f / b	LSTM	To Predict network traffic by a multi-step LSTM model, using the raw data collected from physical channel
47	Wireless Network	Reducing the computing cost of LSTM for NTP	f / b	LSTM	Computational-cost-reduction by means of reconstructing LSTM network in form of a random sparse graph
60	5G	Data privacy protection-aware and accurate prediction with few real data	f / s	LSTM	A NTP model called ctGAN-S2S, based on data augmentation.
20	Cellular Network	Traffic matrix estimation	f / s	LSTM, FFNN, RNN, IRNN, GRU	Analyzing the performance of various RNNs in forecasting TM
61	High-speed networks	Addressing the error propagation	f / b	GPR	Multiple-step-ahead traffic prediction by using a GPR framework which models the traffic at different timescales
62	Internet	QoS, and resource efficiency	n / s	GPR	Online bandwidth allocation method based on GPR
63	Cellular Network	Improving the prediction performance	f / b	LSTM, GP	Two LSTM and GPR modules jointly are adopted to improve the prediction performance
64	Cellular Network	Flow size estimation	f / b	GPR, NN	Estimating the flow size and separating elephant and mice flows using online ML techniques based on historical data as well as online streaming data under hybrid learning settings.
65	CRN	Predicting the near future of spectrum occupancy	n / s	BOL	Modeling the PU channel state indicator sequence as a time series, of which changes are predicted in advance by the proposed BOL model
66	SDN- and NFV-based networks	Cardinality estimation in the field of network measurement	n / b	SGD , RLS, and PA	A sampling-based framework for adaptive cardinality estimation utilizing sampling and online learning
67	Mobile edge network	Minimizing the time-averaged operational cost of MEC	n / s	SGD	A SGD -based faramework for online learning from the changes of the network without needing training data
68	IoT	Real-time feature learning	n / s	ITAE	An incremental deep computation model consisting of two incremental tensor auto-encoders, that is, parameter-based and structure-based incremental learning algorithms
69	LTE	Real-time bandwidth prediction	f / b	Random Forest	A ML framework (called LinkForecast) is developed that utilizes both past throughput and lower-layer information to predict link bandwidth in real time.
70	Cellular network	Modeling the nonlinear dynamics of wireless traffic	f / b	CNN	Treating traffic data as images, and applying the influence degrees of lagged spatial and temporal data to the prediction

(Continues)

TABLE 2 (Continued)

References	Network	Purpose	Tr.	Approach	Description
72	Cellular network	Spatiotemporal modeling and prediction of network traffic	f / s	CNN, LSTM	A compound architecture to extract and applying the spatiotemporal correlation of network data using CNN multi-layer stack, LSTM -based temporal information processing unit, and an attention module
71	Cellular network	Flow size estimation	f / b	DNN	Altering the regression problem of estimating flows bit-rate values to a three-level classifying task, under a semi-streaming fashion setting.
Pure statistical-based NTP solutions					
73	Cellular network	investigating the smoothing factor impact	f / b	S-ESM and ETS-ESM methods	Different varieties of exponential smoothing methods are investigated based on their smoothing factor impact on NTP accuracy.
22	TCP/IP	QoS-aware routing	f / s	ARIMA	The delay of packets is predicted based on the last trend extracted from the queue's behavior on the router
74	Ad hoc IoT networks	Trend change detection	n / s	Online data analysis by dynamic-sliding window	A statistical light-weight Trend Change Detection (TCD) method in an online manner
75	Ad hoc IoT networks	Trend change detection	f / s	Data analysis	Recognizing majore trend's change points in a Poisson distribution dataset through a recursive matrix-based model
76	IoT	Contextual outliers detection	f / s	Data analysis	Fragmenting time-series dataset into segments based on trend change points to discover the contextual outliers
36	Cloud network	Reducing dpendence on historical data	f / s	Fuzzy-ARIMA	The statistical Fuzzy-ARIMA model conducts short-term traffic prediction in real time, based on the historical data blocks provided by a sliding window
77	SDN-based network	QoS-aware routing	n / s	Linear regression	Applying linear regression-based online learning for dynamically predict the latency, in order to updating the flow rules in network devices
Hybrid NTP Solutions					
50	C-RAN	Large-scale NTP in a cost-efficient manner	f / s	GP and ADMM	A C-RAN-based architecture on which, an ADMM and cross-validation empowered scalable GP framework could be performed.
79	SDN-based networks	QoS-aware routing	n / s	LSTM and DRL	Forecasting the future network traffic and detecting the optimal route, dynamically by using a DRL agent.
80	IIoT	Short-term traffic prediction in Real-time	f / s	RL, MCL,Q-learning, and KL	Real-time prediction of short-term time-varying features of network traffic via a RL approach, as well as adopting a dictionary learning algorithm to reduce the complexity of the proposed model
81	Cellular network	forecasting network-wide traffic behavior	f / b	CNN, LSTM, and DWT	Temporal decomposition of the original TM via a wavelet transform, as well as extracting spatial patterns and long-term temporal features correspondingly by CNN, and an input sequence aware LSTM.

(Continues)

TABLE 2 (Continued)

References	Network	Purpose	Tr.	Approach	Description
82	Cellular network	To predict cell-station traffic volumes	f / b	K-means clustering, Wavelet Tramsform, and RNN (Elman NN)	Decomposing the clustered network traffic flows into high and low frequencies to facilitate the prediction process.
83	Cellular network	To improve network traffic prediction by using sequence mining	f / b	LSTM , ANFIS, and FCM	Accelerating the performance of prediction models by using preprocessing and clustering the training data
Comparative works					
78	Cellular networks	Comparison in terms of complexity and accuracy	f / s	GP, SARIMA, and sinusoid superposition	In a 2-fold process, the kernel is designed based on captured traffic patterns. As well as leveraging the Toeplitz structure of the covariance matrix, the computational complexity of hyperparameter learning is decreased.
84	Cellular network	Long- and short-term traffic forecasting	f / b	Auto regression, NN, and GP	Traffic pattern forecasting, based on big data, ML, and network KPIs
85	Wireless network	Decreasing the training overhead without dilution of accuracy	f / b	LSTM , SARIMA	Studying the impact of the training-related factors on accuracy and overhead through camparison of two approaches
43	LTE-A	Comparison of approaches in terms of accuracy and learning time	f / b	LSTM, ARIMA, FFNN	The survey highlights outperforming of proposed LSTM model to FFNN and ARIMA in terms of prediction performance and training time efficiency
86	Wireless network	Real-time NTP	f / b	LV, MA, DES, AR, ARMA, ANN, and Wavelet	A comprehensive comparison, and analyzing three main classes of predictors from various aspects
87	Cellular network	Forecasting TM in large networks	f / s	LSTM and ARIMA	Model parameters optimization to forecast large scale TMs
27	Internet	Comparing various usage schemes of models	f / b	FARIMA, FARIMA/GARCH, and NN	Comparison between distict schemes of applying composed approaches to dynamic bandwidth allocation task
88	Internet	Comparing various usage schemes of models	f / b	FARIMA, NN	FARIMA and MLP models are used in a framework to compare different application methods of traffic prediction, including individual, hybrid, and selective schemes

Abbreviations: ADMM, alternating direction method of multipliers; AR, autoregressive; ARMA, autoregression moving average; ARIMA, Autoregressive integrated moving average; BOL, Bayesian online learning; CNN, convolutional neural network; CRN, cognitive radio networks; DES, double exponential smoothing; DRL, deep reinforcement learning; DNN, Deep Neural Networks; DWT, discrete wavelet transform; ETS-ESM, error, trend, seasonal exponential smoothing method; FARIMA, fractional autoregressive integrated moving average; FCM, fuzzy-C-means; FFNN, feedforward neural network; GARCH, generalized autoregressive conditional heteroskedasticity; GPR, Gaussian process regression; GRU, gated recurrent unit; IoT, Internet of Things; IRNN, identity recurrent neural network; ITAE, incremental tensor auto-encoders; KL, Kullback-Leibler; LSTM, long short-term memory; LV, last value; MV, moving average; NFV, network function virtualization; PA, passive-aggressive; QoS, quality of service; RL, reinforcement learning; RLS, recursive least squares; RNN, recurrent neural network; S-ESM, simple exponential smoothing methods; SARIMA, Seasonal-ARIMA; SDN, software-defined network; SGD, stochastic gradient descent; Tr., Training: Online *(n)* or Offline *(f)* / Batch *(b)* or Stream *(s)*

setting in terms of the adopted learning algorithm, where the Linear Regression model has given way to an LSTM-based model.

Nie et al[80] have modeled the NTP problem in the Intelligent Internet of Things (IIoT) ecosystem as a Markov decision process. In promising to extract short-term time-varying features of network traffic in real-time, and aiming at minimizing the training data size, it proposes a RL-based approach consisting of Monte-Carlo learning (MCL), Q-learning, and Kullback-Leibler (KL) divergence. Moreover, to deal with degrading impact of the vast state space of

Monte-Carlo-Q-learning in IIoT ecosystems, a greedy adaptive dictionary learning algorithm is proposed that reduces the computational complexity. To discover future network-wide traffic behavior, Zhao et al[81] introduce a TM prediction method, coined WSTNet, a complementary combination of CNN and LSTM, and utilizing Discrete Wavelet Transform (DWT) as a feature engineering tool. The method is comprised of three phases. First, at the preprocessing step using DWT, the original TM series is decomposed into multilevel time-frequency subseries at various timescales; second, to draw out the spatial patterns of traffic flows between endpoints, CNN without pooling is leveraged. Finally, the LSTM with self-attention technique is adopted to extract the TM series' long-term temporal dependencies.

Zang et al[82] have composed K-means clustering, Wavelet decomposition, and Elman neural network (ENN) in a framework to predict cell-station traffic volumes by using the spatial-temporal information of cellular traffic flow. After clustering the multiple BS traffic flows, the integrated time series are decomposed into high and low frequencies through Wavelet transform. This creates new subdivisions of data with higher stability and more tractable features facilitating the prediction process.

The main idea of Aldhyani et al[83] is to make the adopted ML models robust by using specific techniques. It proposes a 2-fold process to improve the performance of LSTM and Adaptive neuro fuzzy inference system (ANFIS) models in predicting network traffic. In the first step, the data is preprocessed by using the weighted exponential smoothing model then, the historical data is clustered by non-crisp Fuzzy-C-Means (FCM). The presented results show that identifying and classifying existing patterns of data can improve the performance of LSTM and ANFIS models in forecasting the network traffic behavior.

The third part of Table 2 shows the list of the reviewed literature in this category.

3.4 | Comparative works

Finding a good prediction solution by comparing the characteristics, shortcomings, and strengths of statistical- and ML-based methods has formed one of the areas of interest in researches related to this discipline.

Xu et al[78] have established a wireless traffic prediction model by applying the GP method based on real 4G traffic data. Observation of network traffic to capture periodic trend and dynamic deviations of data, as well as leveraging the Toeplitz structure in the covariance matrix to reduce the computational complexity of hyperparameter learning are the two pillars of the proposed method. The proposed model claims a significant reduction in computational complexity, and high accuracy.

In order to highlight the magnitude of the impact of network Key Performance Indicators (KPIs) on prediction accuracy, a comparative research by Le et al[84] has been conducted comprising two different settings. The first one addresses the time-series traffic forecasting mainly drawn on exploiting the traffic's historical traces, and the second involves with traffic's KPIs prediction through analyzing the relationship between KPIs and future patterns of network traffic. In this analysis, the two main criteria considered are Mutual Information (MI) and Relative Mutual Information (RMI). The performance of three algorithms, including GP, ANN, and AR, has been examined in both settings, reflecting a better performance and accuracy from the GP.

The research by Soheil et al,[85] beyond a solely theoretical basis, addresses some practical aspects of training and deployment of ML models for predicting real-world network data streams in telemetry systems, issues that challenge applying an ML model and achieving the expected accuracy. To this end, various training-related aspects, including the volume, freshness, and selection of training data, have been examined to show their impact on the accuracy and overhead (and thus feasibility) of both adopted models, namely LSTM and SARIMA. Further, utilizing separate models for different segments of a data stream is explored as well. Drawn on the achieved results, the article concludes that any network modeling often needs to be customized based on its target application for a specific data stream from a particular network.

Jaffry[43] introduces a LSTM-based model for traffic prediction in the Long Term Evolution-Advanced (LTE-A) network. It has been compared with similar paradigms based on ARIMA and FFNN in terms of performance and accuracy. The results of this study indicate the superiority of LSTM over FFNN and ARIMA. In addition, the efficiency of LSTM in working with small amounts of training data is another advantage mentioned for this model.

A comprehensive comparison between three predictor classes, focusing on considering some initiative criteria, is proceeded by Faisal et al.[86] The compared techniques are Last Value (LV), Windowed MA, Double Exponential Smoothing (DES), AR, ARMA, ANN-based predictors, and wavelet-based predictors. Besides considering the accuracy, as seen in other similar studies, this study also investigates overhead in terms of both computation cost and power consumption. Further, employing an initiative synthesized metric called Error Energy Score (EE-Score), accuracy and energy consumption have been combined into a single global performance score for comparing predictors. Among the article's conclusions, this point is noteworthy that contextual conditions (ie, network characteristics) determine the proper predictor.

Azzouni and Pujolle[87] have suggested an LSTM framework for predicting TM in large networks. The potency of RNNs for sequence modeling tasks stems from their architectural characteristics. Having a cyclic connection over time enables them to store the activation from each time step in an internal state indicator, which forms a temporal memory. To establish continuous data feeding and learning required for real-time prediction of the current traffic vector Xt, a sliding window technique has been adopted which provides a fixed number of previous time-slots to learn of from. In Reference 27 by Chriskatris and Daskalaki, under different application strategies, including single, hybrid, selective, and combined, three models namely FARIMA, FARIMA/GARCH, and NN and their compositions have been applied to some schemes of dynamic bandwidth allocation for video transmission through network. The comparison outcomes highlight the predictive capability and cost effectiveness of a hybrid model consisted of FARIMA/GARCH, and NN.

In Reference 88 by Chriskatris and Daskalaki, individual, hybrid, and selective usage schemes of models for composing prediction approaches are examined. To this end, FARIMA and MLP models and their hybridization are used, while in the selective scheme, swapping between FARIMA and NN models is proceeded based on the White NN test considering the nonlinearity of traffic data. Outcomes indicate the outperforming of the hybris method.

The last part of Table 2 shows the list of reviewed literature in this category by comparing them in different aspects, for example, network type, purpose and approach.

4 | CHALLENGES OF EXISTING SOLUTIONS IN NTP

Given the inherent complexity of working with big data, especially in network environments, there are many challenging issues that need further investigation. In this section, some open challenges in the field of NTP will be highlighted.

- **Lack of specialized theoretical framework:** The ML methods used in NTP address prediction needs in the general context of time-series-based problem solving. However, in practice, the impact of numerous environmental factors, such as topology, speed, heterogeneity, etc., on the performance of ML models are very significant. In this regard, establishing a dedicated and promising theoretical framework for the application of ML in computer networks is one of the basic needs in this area.
- **Ground truth:** ML techniques are extremely correlated to what is known as Ground-Truth (GT). GT reflects the abundance of labeled data as well as the validity of the labels assigned to the observations. In the main applications of ML, regression, and classification, the GT extracted from the training data is the basis for comparison and inference to estimate the new data labels and, in data categorization, it is used to evaluate the accuracy of model performance. One of the main challenges involved, especially in dynamic environments such as network traffic, is the scarcity of labeled data or delays in accessing labeled data; a situation in which the semi-supervised strategy becomes relevant. PCA techniques, labeling manually or synthetically, or Active-Learning techniques are among the measures that could be taken to overcome the problem.
- **Representative datasets:** Shortage of public and representative datasets to train ML models is a fundamental challenge network-related prediction scenarios.[89] Due to the data-driven nature of ML solutions, lack of proper representative data or poor quality can lead to reduced accuracy. On the other hand, gaining access to required representative data is often subject of strict constrains.[12]
- **Accuracy versus speed:** Since increasing accuracy of ML techniques may lead to higher computational load in some cases as well as time, any improvement in accuracy will come at the cost of slowing down the training of the ML models and vice versa. Finding the equilibrium point between these two parameters and using hybrid methods for NTP while the model is being (re)-trained is recommended in this regard.[12]
- **Retraining:** In dynamic environments, due to constant changes, model retraining is essential to maintain validity. Unlike static environments with a limited specific dataset, allowing model retraining at any time and retraining dynamic models involve complex technical problems. Even if available in terms of access to previous data, the task will not be feasible with conventional static methods due to high computational cost. In online environments such as computer networks, as the acceleration of change increases, the barriers to model retraining become more complex. One common solution is to use some incremental methods for retraining in which the model is only updated with new observations.

- **The *Theory of Networks* challenge**: Introduced by David Meyer in 2017,[12] this concept explains that the trained models by public datasets cannot be used in real-world applications. In this case, each ML model should be trained separately for each network in a real-world application. Considering this challenge, ML methods with very high training time are not appropriate to use in NTP models as the network can not be left unattended for a long period of time.
- **Deployment issues:** In modern prevalent frameworks for software development and support, such as DevOps and DevSecOps, the software life cycle in its traditional form has been fundamentally transformed and influenced by concepts such as CI/CD, delivery flow, workflow, etc. These frameworks would impose specific rules, automations, and restrictions to the operating environment regarding testing and deployment of software updates and changes.[90] As such, deploying online-generic NTP models that during test phase need to access to real-world network data is an important challenge. In such environments, the criteria for evaluation of new updates and deployment licensing is subject to passing a comprehensive test set to ensure that each component and the whole end-to-end workflow is not degraded through new deployments. Due to the dynamic nature of live streaming data and the data-driven structure of online-generic NTP approaches, gaining accurate and reliable results through testing, if not impossible, but is notoriously difficult.[91] Although the classical AutoML accelerates the development of ML solutions, it does not address those issues concerned in ML systems' deployment. Auto-Adaptive Machine Learning (AAML) is a notion composed by[91] to respond to this problem. It proposes an architecture for ML systems lying in the bed of continual learning concept, aiming to address the aforementioned challenge by continuous update of deployed models.

5 | CONCLUSION

Network Traffic Prediction is considered as an important solution to address the various challenges in networking, for example, resource provisioning, congestion detection, and fault tolerance. The field of NTP is still in infancy as different models and techniques are used to predict the network traffic depending on network's type and purpose. Although ML techniques are widely used for NTP, there are some challenges that should be addressed to improve their efficiency in case of NTP, for example high computational cost, training, and retraining difficulties, Intrinsic large volatility of data, etc. On the other hand, statistical-based techniques are used traditionally for network traffic prediction; however they suffer from various challenges, for example, the excessive volume of needed historical data and their inherent constraints to deal with nonstationary time-series. In this paper, we have surveyed both ML- and statistical-based techniques for NTP. Our study shows that most existing solutions are not efficient in real-world applications due to the special characteristics of network data processing in modern networking technologies such as Theory of Networks, retraining challenge and ground truth.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

REFERENCES

1. Shahraki A, Abbasi M, de Piran M, Chen M, Cui S. A comprehensive survey on 6g networks: applications, core services, enabling technologies, and future challenges; 2021. arXiv preprint arXiv:2101.12475.
2. Shahraki A, Taherkordi A, Haugen Ø, Eliassen F. A survey and future directions on clustering: from WSNs to IoT and modern networking paradigms. *IEEE Trans Netw Serv Manag*. 2020;18(2):2242-2274.
3. Shahraki A, Kaffash DK, Haugen O. A review on the effects of IoT and smart cities technologies on urbanism. Proceedings of the 2018 South-Eastern European Design Automation, Computer Engineering, Computer Networks and Society Media Conference (SEEDA_CECNSM); 2018:1-8.
4. Horwitz L. The future of IoT miniguide: the burgeoning IoT market continues Technical report. CISCO, San Jose, CA; 2019.
5. Shahraki A, Geitle M, Haugen Ø. A comparative node evaluation model for highly heterogeneous massive-scale Internet of Things-mist networks. *Trans Emerg Telecommun Technol*. 2020;31(12):e3924.
6. Mohammadi M, Al-Fuqaha A, Sorour S, Guizani M. Deep learning for IoT big data and streaming analytics: a survey. *IEEE Commun Surv Tutor*. 2018;20(4):2923-2960.
7. D'Alconzo A, Drago I, Morichetta A, Mellia M, Casas P. A survey on big data for network traffic monitoring and analysis. *IEEE Trans Netw Serv Manag*. 2019;16(3):800-813. doi:10.1109/TNSM.2019.2933358
8. Abbasi M, Shahraki A, Taherkordi A. Deep learning for network traffic monitoring and analysis (NTMA): a survey. *Comput Commun*. 2021;170:19-41.
9. Kong F, Li J, Jiang B, Zhang T, Song H. Big data-driven machine learning-enabled traffic flow prediction. *Trans Emerg Telecommun Technol*. 2019;30(9):e3482.

10. Shahraki A, Abbasi M, Taherkordi A, Kaosar M. Internet traffic classification using an ensemble of deep convolutional neural networks. Proceedings of the 4th FlexNets Workshop on Flexible Networks Artificial Intelligence Supported Network Flexibility and Agility; 2021:38-43; IEEE.
11. Abbasi M, Shahraki A, Piran MJ, Taherkordi A. Deep reinforcement learning for QoS provisioning at the MAC layer: a survey. *Eng Appl Artif Intell*. 2021;102:104234.
12. Ayoubi S, Limam N, Salahuddin MA, et al. Machine learning for cognitive network management. *IEEE Commun Mag*. 2018;56(1):158-165. doi:10.1109/MCOM.2018.1700560
13. Aroussi S, Mellouk A. Survey on machine learning-based QoE-QoS correlation models. Proceedings of the 2014 International Conference on Computing, Management and Telecommunications, ComManTel 2014:200-204; IEEE.
14. Bui N, Cesana M, Hosseini SA, Liao Q, Malanchini I, Widmer J. A survey of anticipatory mobile networking: context-based classification, prediction methodologies, and optimization techniques. *IEEE Commun Surv Tutor*. 2017;19(3):1790-1821.
15. Jiang M, Wu C-M, Zhang M, Hu D-m. Research on the comparison of time series models for network traffic prediction. *Acta Electron Sin*. 2009;37(11):2353-2358.
16. Oliveira TP, Barbar JS, Soares AS. Computer network traffic prediction: a comparison between traditional and deep learning neural networks. *Int J Big Data Intell*. 2016;3(1):28-37.
17. Yin X, Wu G, Wei J, Shen Y, Qi H, Yin B. A comprehensive survey on traffic prediction; 2020. arXiv preprint arXiv:2004.08555.
18. Malik S, Ullah I, Kim DH, Lee KT. Heuristic and statistical prediction algorithms survey for smart environments. *J Inf Process Syst*. 2020;16(5):1196-1213.
19. Boutaba R, Salahuddin MA, Limam N, et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *J Internet Serv Appl*. 2018;9(1):16. doi:10.1186/s13174-018-0087-2
20. Vinayakumar R, Soman KP, Poornachandran P. Applying deep learning approaches for network traffic prediction. Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI); 2017:2353-2358; IEEE.
21. Nikravesh AY, Ajila SA, Lung CH, Ding W. Mobile network traffic prediction using MLP, MLPWD, and SVM. Proceedings of the 2016 IEEE International Congress on Big Data (BigData Congress); 2016:402-409; IEEE.
22. Rafsanjani MK, Rezaei A, Shahraki A, Saeid AB. QARIMA: a new approach to prediction in queue theory. *Appl Math Comput*. 2014;244:514-525.
23. Montgomery DC, Jennings CL, Kulahci M. *Introduction to Time Series Analysis and Forecasting*. Hoboken, NJ: John Wiley & Sons; 2015.
24. Krishnaswamy N, Kiran M, Singh K, Mohammed B. Data-driven learning to predict wan network traffic. Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics; 2020:11-18; ACM, New York, NY.
25. Medhn S, Seifu B, Salem A, Hailemariam D. Mobile data traffic forecasting in UMTS networks based on SARIMA model: the case of Addis Ababa, Ethiopia. 2017 IEEE AFRICON: Science, Technology and Innovation for Africa, AFRICON 2017; 2017:285-290. 10.1109/AFRCON.2017.8095496
26. AsSadhan B, Zeb K, Al-Muhtadi J, Alshebeili S. Anomaly detection based on LRD behavior analysis of decomposed control and data planes network traffic using SOSS and FARIMA models. *IEEE Access*. 2017;5:13501-13519. doi:10.1109/ACCESS.2017.2689001
27. Katris C, Daskalaki S. Dynamic bandwidth allocation for video traffic using FARIMA-based forecasting models. *J Netw Syst Manag*. 2018;27(1):39-65. doi:10.1007/S10922-018-9456-1
28. Dengen N. Comparison of SARIMA, NARX and BPNN models in forecasting time series data of network traffic. Proceedings of the 2016 2nd International Conference on Science in Information Technology (ICSITech); 2016:264-269.
29. Hong WC. Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting. *Neural Comput Appl*. 2012;21(3):583-593.
30. Biernacki A. Improving quality of adaptive video by traffic prediction with (F)ARIMA models. *J Commun Netw*. 2017;19(5):521-530. doi:10.1109/JCN.2017.000083
31. Nielsen A. *Practical Time Series Analysis: Prediction with Statistics and Machine Learning*. Sebastopol, CA: O'Reilly Media, Inc.; 2019.
32. Andrysiak T, Saganowski L, Maszewski M, Marchewka A. Detection of network attacks using hybrid ARIMA-GARCH model. *Adv Intell Syst Comput*. 2018;582:1-12.
33. Engle R. GARCH 101: the use of ARCH/GARCH models in applied econometrics. *J Econ Perspect*. 2001;15:157-168. doi:10.1257/JEP.15.4.157
34. Bollerslev T. Generalized autoregressive conditional heteroskedasticity. *J Econ*. 1986;31(3):307-327. doi:10.1016/0304-4076(86)90063-1
35. Chen H, Zhang J, Tao Y, Tan F. Asymmetric GARCH type models for asymmetric volatility characteristics analysis and wind power forecasting. *Protect Control Modern Power Syst*. 2019;4:1-11. doi:10.1186/S41601-019-0146-0
36. Mehdi H, Pooranian Z, Vinueza Naranjo PG. Cloud traffic prediction based on fuzzy ARIMA model with low dependence on historical data. *Trans Emerg Telecommun Technol*. 2019;e3731. doi:10.1002/ett.3731
37. Hajirahimi Z, Khashei M. Hybrid structures in time series modeling and forecasting: a review. *Eng Appl Artif Intell*. 2019;86:83-106. doi:10.1016/j.engappai.2019.08.018
38. Le Duc T, Leiva RG, Casari P, Östberg PO. Machine learning methods for reliable resource provisioning in edge-cloud computing: a survey. *ACM Comput Surv*. 2019;52(5):1–39. doi:10.1145/3341145
39. Alsheikh MA, Niyato D, Lin S, Tan H-P, Han Z. Mobile big data analytics using deep learning and apache spark. *IEEE Netw*. 2016;30(3):22-29.
40. Shahraki A, Abbasi M, Taherkordi A, Jurcut AD. Active learning for network traffic classification: a technical study. *IEEE Trans Cognit Commun Netw*. 2021.

41. Yousefpour A, Fung C, Nguyen T, et al. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J Syst Archit*. 2019;98:289-330.
42. Park J, Samarakoon S, Bennis M, Debbah M. Wireless network intelligence at the edge. *Proc IEEE*. 2019;107(11):2204-2239. doi:10.1109/JPROC.2019.2941458
43. Jaffry S. Cellular traffic prediction with recurrent neural network; 2020. arXiv preprint arXiv:2003.02807.
44. Chen M, Challita U, Saad W, Yin C, Debbah M. Artificial neural networks-based machine learning for wireless networks: a tutorial. *IEEE Commun Surv Tutor*. 2019;21(4):3039-3071. doi:10.1109/COMST.2019.2926625
45. Zhang X, You J. A gated dilated causal convolution based encoder-decoder for network traffic forecasting. *IEEE Access*. 2020;8:6087-6097. doi:10.1109/ACCESS.2019.2963449
46. Trinh HD, Giupponi L, Dini P. Mobile traffic prediction from raw data using LSTM networks. Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC; 2018:1827-1832.
47. Hua Y, Zhao Z, Li R, Chen X, Liu Z, Zhang H. Deep learning with long short-term memory for time series prediction. *IEEE Commun Mag*. 2019;57(6):114-119. doi:10.1109/MCOM.2019.1800155
48. He Q, Moayyedi A, Dan G, Koudouridis GP, Tengkvist P. A meta-learning scheme for adaptive short-term network traffic prediction. *IEEE J Select Areas Commun*. 2020;38(10):2271–2283. doi:10.1109/JSAC.2020.3000408
49. Wen J, Sheng M, Li J, Huang K. Assisting intelligent wireless networks with traffic prediction: exploring and exploiting predictive causality in wireless traffic. *IEEE Commun Mag*. 2020;58(6):26-31. doi:10.1109/MCOM.001.1900211
50. Xu Y, Yin F, Xu W, Lin J, Cui S. Wireless traffic prediction with scalable Gaussian process: framework, algorithms, and verification. *IEEE J Select Areas Commun*. 2019;37(6):1291-1306. doi:10.1109/JSAC.2019.2904330
51. Watkins Christopher JCH, Dayan P. Technical note: Q-learning. *Mach Learn*. 1992;8(3):279-292. doi:10.1023/A:1022676722315
52. Xie J, Yu FR, Huang T, et al. A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges. *IEEE Commun Surv Tutor*. 2018;21(1):393-430.
53. Kulin M, Kazaz T, De Poorter E, Moerman I. A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer. *Electronics*. 2021;10(3):318.
54. Xu Z-X, Cao L, Chen X-L, Li C-X, Zhang Y-L, Lai J. Deep reinforcement learning with sarsa and q-learning: a hybrid approach. *IEICE Trans Inf Syst*. 2018;101(9):2315-2322.
55. Gepperth A, Hammer B. Incremental learning algorithms and applications. Proceedings of the European symposium on artificial neural networks (ESANN), ESANN; 2016.
56. Kuhn M, Johnson K. *Applied Predictive Modeling*. New York, NY: Springer; 2013.
57. Narudin FA, Feizollah A, Anuar NB, Gani A. Evaluation of machine learning classifiers for mobile malware detection. *Soft Comput*. 2016;20(1):343-357.
58. Abbasi M, Shahraki A, Barzegar HR, Pahl C. Synchronization techniques in "device to device-and vehicle to vehicle-enabled" cellular networks: a survey. *Comput Electr Eng*. 2021;90:106955.
59. Alawe I, Ksentini A, Hadjadj-Aoul Y, Bertin P. Improving traffic forecasting for 5G core network scalability: a machine learning approach. *IEEE Netw*. 2018;32(6):42-49. doi:10.1109/MNET.2018.1800104
60. Wang Z, Hu J, Min G, Zhao Z, Wang J. Data augmentation based cellular traffic prediction in edge computing enabled smart city. *IEEE Trans Ind Inform*. 2020;17(6):4179–4187. doi:10.1109/tii.2020.3009159
61. Bayati A, Nguyen KK, Cheriet M. Multiple-step-ahead traffic prediction in high-speed networks. *IEEE Commun Lett*. 2018;22(12):2447-2450. doi:10.1109/LCOMM.2018.2875747
62. Kim J, Hwang G. Prediction based efficient online bandwidth allocation method. *IEEE Commun Lett*. 2019;23(12):2330-2334. doi:10.1109/LCOMM.2019.2947895
63. Wang W, Zhou C, He H, Wu W, Zhuang W, Shen XS. Cellular traffic load prediction with LSTM and Gaussian process regression. Proceedings of the IEEE International Conference on Communications; 2002; IEEE Institute of Electrical and Electronics Engineers Inc.
64. Poupart P, Chen Z, Jaini P, et al. Online flow size prediction for improved network routing. Proceedings of the International Conference on Network Protocols, ICNP; 2016; IEEE.
65. Mikaeil AM. Bayesian online learning-based spectrum occupancy prediction in cognitive radio networks. *ITU J ICT Discov*. 2017;(1):1–6.
66. Cohen R, Nezri Y. Cardinality estimation in a virtualized network device using online machine learning. *IEEE/ACM Trans Netw*. 2019;27(5):2098-2110. doi:10.1109/TNET.2019.2940705
67. Cui Q, Gong Z, Ni W, et al. Stochastic online learning for mobile edge computing: learning from changes. *IEEE Commun Mag*. 2019;57(3):63-69. doi:10.1109/MCOM.2019.1800644
68. Zhang Q, Yang LT, Chen Z, Li P. Incremental deep computation model for wireless big data feature learning. *IEEE Trans Big Data*. 2019;6(2):248-257. doi:10.1109/tbdata.2019.2903092
69. Yue C, Jin R, Suh K, Qin Y, Wang B, Wei W. LinkForecast: cellular link bandwidth prediction in LTE networks. *IEEE Trans Mob Comput*. 2018;17(7):1582-1594. doi:10.1109/TMC.2017.2756937
70. Zhang C, Zhang H, Yuan D, Zhang M. Citywide cellular traffic prediction based on densely connected convolutional neural networks. *IEEE Commun Lett*. 2018;22(8):1656-1659. doi:10.1109/LCOMM.2018.2841832
71. Pfülb B, Hardegen C, Gepperth A, Rieger S. A study of deep learning for network traffic data forecasting. Proceedings of the International Conference on Artificial Neural Networks; 2019:497-512; Springer, New York, NY.
72. Li M, Wang Y, Wang Z, Zheng H. A deep learning method based on an attention mechanism for wireless network traffic prediction. *Ad Hoc Netw*. 2020;107:102258. doi:10.1016/J.ADHOC.2020.102258

73. Tran QT, Hao L, Trinh QK. Cellular network traffic prediction using exponential smoothing methods. *J Inf Commun Technol*. 2019;18(1):1-18. doi:10.32890/jict2019.18.1.1
74. Shahraki A, Taherkordi A, Haugen Ø. TONTA: trend-based online network traffic analysis in ad-hoc IoT networks. *Comput Netw*. 2021;194:108125.
75. Shahraki A, Taherzadeh H, Haugen Ø. Last significant trend change detection method for offline Poisson distribution datasets. Proceedings of the 2017 International Symposium on Networks, Computers and Communications (ISNCC); 2017:1-7; IEEE.
76. Shahraki A, Haugen Ø. An outlier detection method to improve gathered datasets for network behavior analysis in IoT. *Journal of Communications*. 2019;14(6):455-462.
77. Bouzidi EH, Luong DH, Outtagarts A, Hebbar A, Langar R. Online-based learning for predictive network latency in software-defined networks. Proceedings of the 2018 IEEE Global Communications Conference, GLOBECOM 2018 – Proceedings; 2018. IEEEInstitute of Electrical and Electronics Engineers Inc.
78. Xu Y, Xu W, Yin F, Lin J, Cui S. High-accuracy wireless traffic prediction: A GP-based machine learning approach. Proceedings of the 2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings; 2017:1-6; IEEE.
79. Bouzidi EH, Outtagarts A, Langar R. Deep reinforcement learning application for network latency management in software defined networks. Proceedings of the 2019 IEEE Global Communications Conference, GLOBECOM 2019 – Proceedings; 2019; IEEE.
80. Nie L, Ning Z, Obaidat M, et al. A reinforcement learning-based network traffic prediction mechanism in intelligent Internet of Things. *IEEE Trans Ind Inform*. 2020;17(3):2169-2180. doi:10.1109/tii.2020.3004232
81. Zhao JJ, Qu H, Zhao JJ, Jiang D. Spatiotemporal traffic matrix prediction: a deep learning approach with wavelet multiscale analysis. *Trans Emerg Telecommun Technol*. 2019;30(12):e3640. doi:10.1002/ett.3640
82. Zang Y, Ni F, Feng Z, Cui S, Ding Z. Wavelet transform processing for cellular traffic prediction in machine learning networks. Proceedings of the 2015 IEEE China Summit and International Conference on Signal and Information Processing, China SIP 2015 – Proceedings; 2015:458-462IEEE Institute of Electrical and Electronics Engineers Inc.
83. Aldhyani THH, Alrasheedi M, Alqarni AA, Alzahrani MY, Bamhdi AM. Intelligent hybrid model to enhance time series models for predicting network traffic. *IEEE Access*. 2020;8:130431-130451. doi:10.1109/access.2020.3009169
84. Le LV, Sinh D, Tung LP, Lin BSP. A practical model for traffic forecasting based on big data, machine-learning, and network KPIs. Proceedings of the CCNC 2018 - 2018 15th IEEE Annual Consumer Communications and Networking Conference; 2018:1-4.
85. Soheil J, Hammoudeh Z, Durairajan R, Lowd D, Rejaie R, Willinger W. On the practicality of learning models for network telemetry. Proceedings of Network Traffic Measurement and Analysis ConferenceNational Science Foundation; 2020.
86. Faisal IM, Zahid M, Habib D, et al. Efficient prediction of network traffic for real-time applications. *J Comput Netw Commun*. 2019;2019:4067135. doi:10.1155/2019/4067135
87. Azzouni Abdelhadi, Pujolle Guy. A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv preprint arXiv:1705.05690. 2017;.
88. Katris C, Daskalaki S. Comparing forecasting approaches for internet traffic. *Expert Syst Appl*. 2015;42(21):8172-8183. doi:10.1016/J.ESWA.2015.06.029
89. Shahraki A, Abbasi M, Haugen Ø. Boosting algorithms for network intrusion detection: a comparative evaluation of real AdaBoost, gentle AdaBoost and modest AdaBoost. *Eng Appl Artif Intell*. 2020;94:103770.
90. Lohrasbinasab I, Acharya PB, Colomo-Palacios R. BizDevOps: a multivocal literature review. lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2020:12254; LNCS:698-713. 10.1007/978-3-030-58817-5_50
91. Diethe T, Borchert T, Thereska E, Balle B, Lawrence N. Continual learning in practice; 2019. arXiv preprint arXiv:1903.05202.

How to cite this article: Lohrasbinasab I, Shahraki A, Taherkordi A, Delia Jurcut A. From statistical- to machine learning-based network traffic prediction. *Trans Emerging Tel Tech*. 2021;e4394. doi: 10.1002/ett.4394