

Deep Learning for Predicting Traffic in V2X Networks

Ali R. Abdellah ^{1,*} , Ammar Muthanna ² , Mohamed H. Essai ¹  and Andrey Koucheryavy ²¹ Electrical Engineering Department, Al-Azhar University, Qena 83513, Egypt² Department of Telecommunication Networks and Data Transmission, The Bonch-Bruевич Saint-Petersburg State University of Telecommunications, 193232 St. Petersburg, Russia

* Correspondence: alirefae@azhar.edu.eg

Abstract: Artificial intelligence (AI) is capable of addressing the complexities and difficulties of fifth-generation (5G) mobile networks and beyond. In this paradigm, it is important to predict network metrics to meet future network requirements. Vehicle-to-everything (V2X) networks are promising wireless communication methods where traffic information exchange in an intelligent transportation system (ITS) still faces challenges, such as V2X communication congestion when many vehicles suddenly appear in an area. In this paper, a deep learning algorithm (DL) based on the unidirectional long short-term memory (LSTM) model is proposed to predict traffic in V2X networks. The prediction problems are studied in different cases depending on the number of packets sent per second. The prediction accuracy is measured in terms of root-mean-square error (RMSE), mean absolute percentage error (MAPE), and processing time.

Keywords: 5G; V2X; deep learning; artificial intelligence; LSTM



Citation: Abdellah, A.R.; Muthanna, A.; Essai, M.H.; Koucheryavy, A. Deep Learning for Predicting Traffic in V2X Networks. *Appl. Sci.* **2022**, *12*, 10030. <https://doi.org/10.3390/app121910030>

Academic Editor: Vincent A. Cicirello

Received: 30 August 2022

Accepted: 30 September 2022

Published: 6 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In telecommunications, 5G networks and beyond have high requirements in terms of delays, security, reliability of connections, etc. As the number of devices in the telecommunications network increases, leading to a large increase in information flow, the challenge is to find more efficient ways to manage and control the network. As the number of devices in the telecommunications network increases, leading to an extensive increase in the flow of information, the challenge is to find more efficient ways to manage and control the network. Reliable, intelligent methods are required for 5G for adapting network protocols and resource management for different facilities in various environments. Artificial intelligence (AI), defined as any method or system that recognizes its environment and performs actions that increase the chances of success in achieving a particular goal, is a practical solution for developing new complicated communication systems [1,2].

All aspects that make machines smarter belong to the broad field of AI. An AI system that can learn autonomously based on an algorithm is called machine learning (ML). ML is an advanced subfield of AI that produces intelligent systems that continuously learn and become smarter without human intervention. Since intelligent behavior requires extensive knowledge, ML is used in most existing AI applications. ML then uses these data to provide intelligent information or predict a specific outcome when new entries are introduced. ML is well-suited for solving complex problems such as optimizing wireless networks or detecting attacks on the network. Large datasets can be processed using an ML technique called deep learning (DL).

DL has recently gained popularity because it can quickly solve some types of difficult problems in numerous applications and can use both supervised and unsupervised learning algorithms. It depends on multilayer neural networks (MLNN) and methods that handle extensive datasets, and some of these methods outperform traditional artificial neural networks (ANNs) in processing the previous information. Current DL advances promise to solve highly complex, previously unsolvable problems. DL algorithms are important for

predicting traffic behavior because they can learn from large amounts of data and identify patterns more accurately than other methods. Potential traffic predictions enable solutions to improve quality of service (QoS) before it drops. Because DL uses historical data to improve its decision making and achieve higher accuracy, it can be used for predictive analytics. A more secure network is enabled by predicting a potential massive occurrence of data streams at unusual times, which can be best described as data stream-based intrusion. Predicting such large data streams can also eliminate the risk that could affect the performance of the IoT system [3–5].

Internet of Things (IoT) solutions for infrastructure, applications, and security are creating new experiences and enabling more advanced operations based on acquired data at the intelligent edge. The IoT enables connectivity between daily things via sensors, and information can be collected and shared over the Internet. As the number of IoT devices increases, leading to an increase in applications, the amount of data collected will also increase, and IoT services will be developed for operators in various sectors. IoT systems connect with other devices and collect huge amounts of data every day. In addition, IoT systems can be programmed to start specific activities depending on predefined conditions or reactions to collected data from various applications. Moreover, human intervention is required to investigate the aggregated information, produce sensitive information, and create intelligent implementations. Furthermore, they can make decisions and learn from the data they collect. For advanced automated deployments, smart IoT devices must also be equipped with the capabilities to allocate resources, communicate, and provide network services [6,7].

Recently, it has become possible to use AI technology in 5G mobile networks to optimally design the physical layer, make complex decisions, manage the network, and optimize resources in such networks. Moreover, the developing big data technique provides an outstanding opportunity to learn the critical features of wireless networks and gain a more accurate and deeper understanding of the performance of 5G cellular networks [8,9].

Nowadays, ML is successfully used in a variety of fields, as it can process massive amounts of data and discover dependencies that are difficult for humans to identify manually. Moreover, such an algorithm can determine the optimal solution to a given problem by statistically analyzing a large amount of data. For example, ML algorithms are used to predict accidents in heat supply systems. ML reduces the number of dangerous events (e.g., the failure of a rail on a railway track) by predicting rare, dangerous failures based on big data processing. In addition, they can be successfully used for delay-sensitive applications such as medical care, safety, and contingency responses such as remote intensive patient care. A significant event needs to be reported to a monitoring organization during a specific period to decide on suitable actions. In addition, the algorithms of ML are used to solve problems in antivirus scanning, hydrocarbon reservoir exploration, business, and so on [10–12].

The following are some of the main motivations behind this study:

- ✓ Improve QoS demands and network monitoring for resource management and security.
- ✓ Monitor network availability and activity for anomaly detection, including security and operational issues.
- ✓ We proposed a DL-based LSTM method for the following reasons:
 - a. It can predict future time series data more accurately than traditional time series models and stores historical series data over a long period.
 - b. It adjusts data faster and more efficiently than traditional time series models.
 - c. Better performance when processing larger datasets.
 - d. Another advantage over other approaches to time series forecasting is that it maximizes the accuracy of the learning method across training iterations.
 - e. The more data added to the model, the smarter and better it can estimate traffic volumes, which is important for real-time traffic forecasting.
- ✓ Lack of awareness of the parameters of IoT communication.

- ✓ Insufficient machine learning analysis prevents adequate prediction accuracy from being achieved.
 - ✓ The computational complexity of QoS measurement.
- The main contributions to the proposed work are summarized as follows:
- The unidirectional long short-term memory (LSTM) with a DL for traffic prediction in the vehicle-to-everything (V2X) network was proposed.
 - The model DL was trained in different cases, based on the number of packets sent per second, with unidirectional LSTM in a V2X environment. The objective was to determine the best case that would offer us the best accuracy.
 - In terms of prediction accuracy, a comparison was made between the different cases depending on the packets transmitted per second: 4 packets/s, 6 packets/s, 8 packets/s, 10 packets/s, 12 packets/s, and 14 packets/s.
 - Prediction accuracy was determined using root-mean-square error (RMSE), mean absolute percentage error (MAPE), and processing time, using MSE as the loss function and a learning rate of 0.1.
 - Finally, the simulation findings demonstrate the following:
 - ✓ The best prediction accuracy was found for the number of packets transmitted of 4 packets/s, which outperforms the competitors and shows outstanding performance.
 - ✓ On the other hand, the prediction accuracy when using 14 packets/s is lower than the other models.
 - ✓ The model with a prediction of 12 packets/s has the fastest processing time, but the models with a prediction of 14 packets/s have the slowest processing time compared to their competitors.

The paper is organized as follows: Section 2 reviews the relevant literature; Section 3 introduces V2X simulation; Section 4 introduces LSTM with deep learning; Section 4 discusses training DL with LSTM; Section 5 discusses simulation results; finally, Section 5 concludes the paper.

2. Literature Review

Several studies have addressed the prediction of traffic in 5G mobile networks using ML. In this work, we focus on predicting V2X traffic using the DL method based on the LSTM model. Therefore, in this section, we provide an overview of previous studies related to our main topic.

Gao [13] investigated the prediction accuracy of the 5G cellular network and developed a smoothed LSTM traffic forecasting method. The model adjusts the number of hidden layers and units under the adaptive prediction accuracy approach to minimize the randomness of the 5G traffic sequence, simultaneously. Selvamanju et al. [14] comprehensively overviewed the current ML methods for predicting cellular traffic information in 5G networks. Trinh et al. [15] addressed cellular traffic and performed traffic prediction using an LSTM model with recurrent neural networks (RNNs). The information about mobile traffic is obtained from the physical downlink control channel (PDCCH). They evaluated the proposed method's single-step and long-term prediction errors considering different periods with observed values. Chakraborty et al. [16] developed a strategy to incorporate analytical time series into the 5G core and predict risks that could lead to system failures.

Abdellah et al. [17] used a DL method built on an LSTM network to implement a time series prediction of energy consumption for drone-based MEC. Four cases were examined to determine the accuracy based on the training's learning rate. They assessed the prediction accuracy using RMSE and MAPE as measures of prediction accuracy to identify the best prediction accuracy and the maximum average improvement in prediction accuracy.

Zhu et al. [18] presented an intelligent base station sleep system (BS) to reduce system power consumption while guaranteeing the best QoE. In addition, they introduced an LSTM learning approach for predicting the traffic allocation in the service area to determine when to trigger the BS sleep process; in addition, they created a successful three-step

process to choose which of the BSs should be awakened or put to sleep. In [19], traffic prediction was performed using DL and an LSTM network to make predictions about traffic activity in IoT-enabled edge computing.

Wang et al. [20] presented a spatiotemporal analysis of mobile network traffic and gave an overview of current research in this area. They proposed a graph-attention network based on time series similarity. Network traffic prediction (NTP) based on a DL using an LSTM network was investigated in [21,22]. Zhou et al. [23] addressed the short-term forecasting of 5G traffic flow using edge computing for efficient smart city services. A new framework for predicting traffic flows based on the LSTM network, focusing mainly on these irregular traffic flows, has been proposed in [24].

The authors of [25] investigate the validity and feasibility of RNNs, and in particular, LSTMs as potential tools for such predictions. Thus, a new method of QoS prediction is presented that uses an LSTM model and network-based QoS metrics to identify patterns and predict QoS for connected and automated vehicles (CAV) that will soon be available. Guerra-Gómez et al. [26] suggested and analyzed a new resource demand prediction strategy using an ML algorithm. The support vector machine (SVM), time-delay neural network (TDNN), and LSTM were investigated and compared to determine the optimal prediction method. Chaalal et al. [27] investigated a prediction of base station mobility to extend service in 5G networks. A DL using an LSTM network with adjusted hyperparameters is proposed to predict the short-term traffic speed on a parallel multilane arterial road in a growing country such as Vietnam [28]. For cellular traffic prediction, Fawaz et al. [29] proposed a model combining a single-exponential smoothing LSTM. The single-exponential smoothing method was used to fit the volume due to the complexity and diversity of network traffic forms. An LSTM model was applied to the output of a single-exponential model to predict network utilization. The smart system was estimated against actual mobile network traffic collected in a Kaggle dataset.

3. V2X Simulation

To further model the V2X network in the smart city, the MATLAB computing platform was used. First, a mobility model for the V2X network was developed. Using a virtual mobility map, the ad hoc on-demand distance vector (AODV) routing protocol was created to investigate and evaluate its performance. The road network was developed through the creation of mobility maps that include basic entities such as city size, nodes, and RSUs (roadside units). The size of the city is needed to determine the boundaries in which the nodes of the V2X network move in random directions for the implementation of AODV. To implement the AODV routing protocol, the maximum size of the city and the number of nodes are required, and multiple RSUs must be set. If the size of the city is greater than the required simulation period, it is automatically enlarged. It is assumed that the size of the city is 100×100 on the x-y axes. In the mobility model, the nodes on the city boundary can move along fixed routes in any direction.

The simulation of a V2X in a smart city is shown in Figure 1. In Figure 1., the points represent individual nodes and locations of the RSU, characterized by identification numbers corresponding to the structure and configuration of the network. Nodes 20 and 70 serve as the model's starting point and closing point, respectively. The simulation module chooses the simulation's start and finish times and visualizes the network architecture. Nodes that move randomly can connect to other nodes that are far away from them due to the position of the RSU on the simulation map. The RSU makes it possible to connect to moving cars so that messages such as traffic reports and safety alerts can be transmitted.

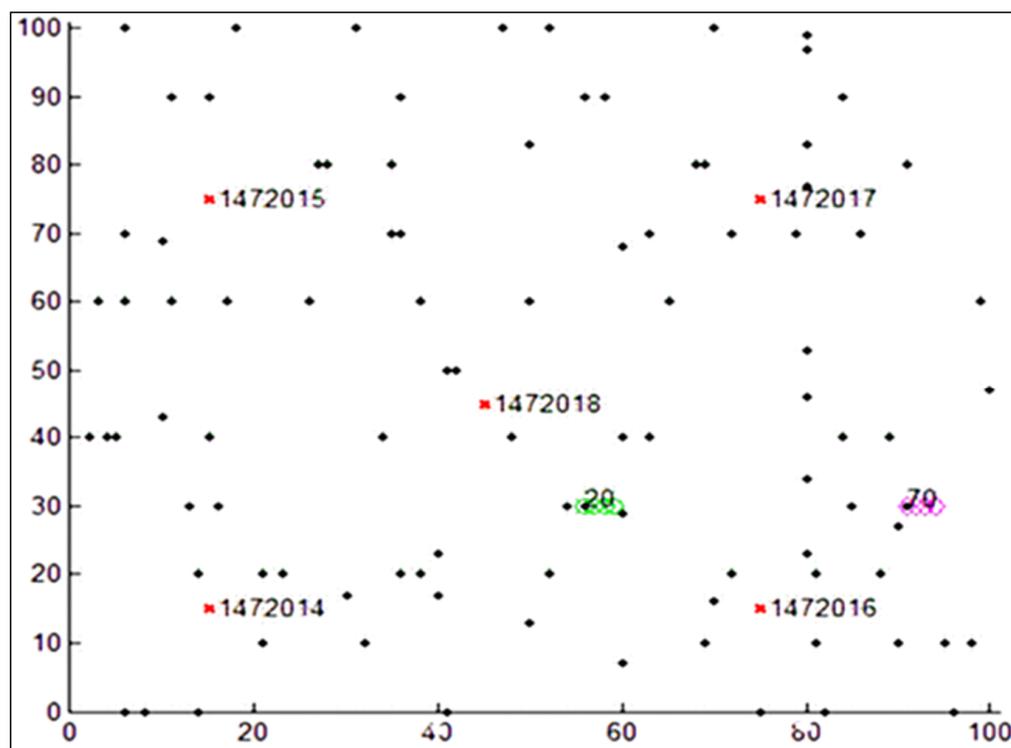


Figure 1. V2X simulation using MATLAB.

4. LSTM Deep Learning

Many techniques are needed to increase the accuracy of traffic feature prediction to solve the problems of 5G mobile networks and not further reduce the effectiveness of the QoS system. So far, numerous ML methods have been developed to optimize the accuracy of traffic forecasts. One of the essential methods is deep learning (DL), a particular form of MLNN. ANNs are characterized by a series of networks within the loop and are more potent than conventional ANNs. Each net within the loop receives signals from the preceding net, executes a series of procedures, generates output data, and transmits information to the succeeding network.

Conventional RNNs are unsuitable for cases where each object must be “remembered” over a long period; they are delayed when there is no link between past and future information. The influence of a hidden state or an input with a step t on the following states of the feedback network decreases rapidly. Among the solutions recently applied to DL, the most important is the modification and complexity of the configuration of the “building block” of the RNN. It becomes clear that instead of a single number influenced by all the following states, we can build a specific kind of cell wherein we model a “long-term memory” in one form or another, with the procedures starting from this “memory cell”. These cells would have more than one set of weights; as with a typical neuron, learning becomes more challenging, but it is usually beneficial in training. Some applications require new data, while others require more historical data.

A method for predicting events over time is called time series forecasting. By examining historical trends and assuming that future trends will be similar to historical trends, future events are predicted. It is used in many areas of science in various applications, such as control engineering, pattern recognition, resource allocation, signal processing, statistics, and weather forecasting. I/O time series rely on forecasting using models fitted to historical data to predict future observations over a period of time. The goal of time series forecasting is to predict future values over a period of time. It involves building models based on past data, drawing conclusions from them, and using them as a basis for future strategic decisions.

A type of RNN called an LSTM network [17,19,21,22,30] uses past inputs to predict the future output. LSTMs were developed primarily to address the problem of long-term dependence. LSTMs are robust in the time-series forecast as they can memorize historical information. This is important in our case because past V2X traffic is crucial for predicting future traffic data. LSTM outperforms conventional time series methods in improving prediction. It is also more competent and rapid at fitting data than conventional models. They also handle massive amounts of data better than traditional time series models. Long-term memory is almost their hypothetical behavior, not a thing they aim to make. A conventional recurrent neural network (RNN) contains a sequence of repeating units with a simple structure, e.g., a hyperbolic tangent layer (). The LSTM network’s structure is depicted in Figure 2.

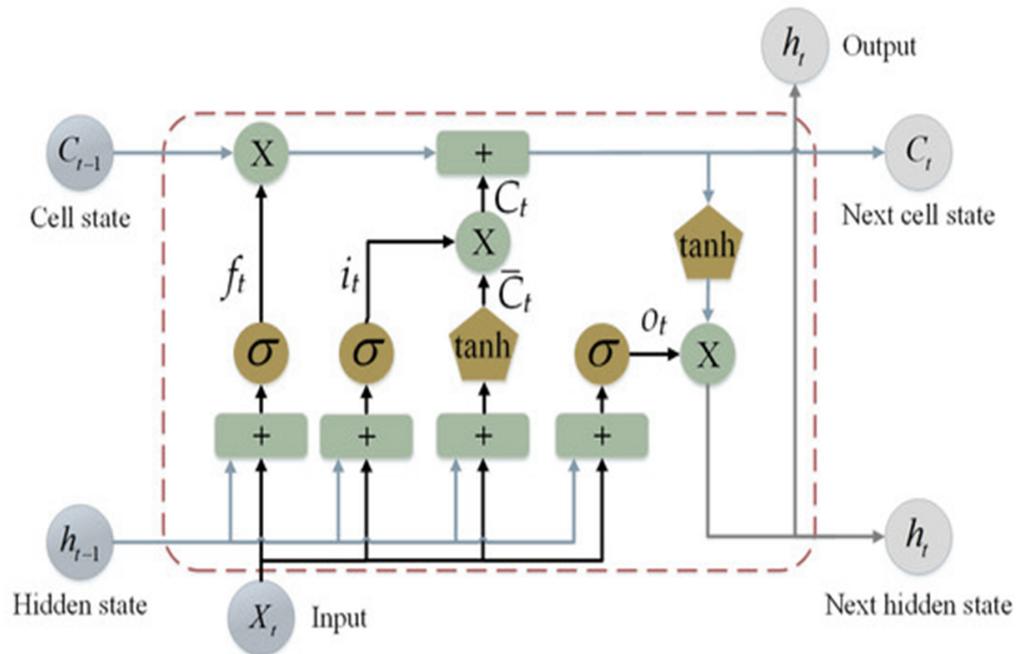


Figure 2. Structure of LSTM Model.

The cell state, which runs across the upper part of the model, is the LSTM’s main feature. With only a few serial correlations, it goes directly through the sequence, and the signals can be rotated without change. The gate structures allow the LSTM to remove or add data from the cell state, and these gates carefully control this capability. They consist of pointwise multiplication and the ANN sigmoidal layer (Equation (1)). The interval (0,1) limits the sigmoidal layer output to specify the components that have passed through the gate. There are three gates in the LSTM to protect and regulate the cell state. The output of the LSTM gates is equal to 0 or 1 since they have sigmoidal activation functions (Equation (1)).

$$\sigma(x) = 1/(1 + e^{-x}) \tag{1}$$

Firstly, LSTM determines any signals to forget from the cell state via a sigmoidal layer known as the forget gate. It takes h and x as inputs, and its outputs have values from 0 to 1, where “1” means “totally keep” and “0” means “totally delete”.

$$f_t = \sigma(W_f(h_{t-1}, x_t) + b_f) \tag{2}$$

Next, we must find out what new signals are stored in the cell state. This process can be divided into two parts. First, the input gate layer of the sigmoidal layer determines which values need to be updated. Then, the candidate cell state is created by the \tanh layer

using a vector of the most recent candidate values, C . In the following step, these two values are combined to update the state.

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

$$\tilde{C}_t = \tanh(W_c(h_{t-1}, x_t) + b_c) \tag{4}$$

Input x at time t and a hidden state h at time $t - 1$ to determine the new information that will be passed to the cell state. \tanh is the activation function in this case. The value of the new information will range from -1 to 1 due to the \tanh function.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{5}$$

The state of the past cell to be updated is multiplied by f , losing the information we forgot. Then $i * C$ is added. These are the current candidate values, estimated according to how we want to update each state variable.

$$C_t = f_t \times C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

In the end, we must specify everything we want to obtain as an output. Most importantly, apply the sigmoid layer and determine which parts of the cell state should be taken out (the output). Then, apply the cell state over \tanh (to put all values in the interval $[-1, 1]$) and multiply it by the sigmoid gate output to produce a hidden state.

$$o_t = (W_o(h_{t-1}, x_t) + b_o) \tag{7}$$

$$h_t = o_t \times \tanh(C_t) \tag{8}$$

Table 1 shows the list of LSTM variables.

Table 1. List of LSTM variables.

$x_t \in \mathbb{R}^d$	Input vector to the LSTM unit
$f_t \in (0, 1)^h$	Forget gate's activation vector
$i_t \in (0, 1)^h$	Input/update gate's activation vector
$o_t \in (0, 1)^h$	Output gate's activation vector
$h_t \in (-1, 1)^h$	The hidden state vector is also known as the output vector of the LSTM unit
$\tilde{C}_t \in (-1, 1)^h$	The candidate cell state is also known as the cell input activation vector
$C_t \in \mathbb{R}^h$	Cell state vector
$C_{t-1} \in \mathbb{R}^{h-1}$	The previous cell state vector
$W_x \in \mathbb{R}^{h \times d}, W_h \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$	Weight matrices and bias vector parameters must be learned during training
σ	Sigmoid function
\tanh	Hyperbolic tangent function

The superscript letters d and h denote the number of input features and the number of hidden units, respectively.

5. Training DL-Based LSTM

DL-based LSTM cells can be trained using a set of training sequences by combining a learning algorithm, such as gradient descent, with backpropagation in computing the gradients required in the learning process to change each LSTM network weight in proportion to the derivative of the error (on the LSTM network's output layer) with the corresponding weight. The issue with gradient descent in RNN is that as the distance between real events grows, the error gradients quickly vanish. Before one can realize the importance of connecting

remote inputs, recurrent networks attempt to connect the result to the events in several steps. The error is still present in the LSTM unit cell even if the error values propagate back from the output layer. All LSTM cells receive this error repeatedly until they learn to ignore it.

To predict the input and output time series, the future value of one time series must be predicted based on another time series. To predict the time series, either the historical values of both time series (for more accuracy) or simply one time series (for a simpler technique), can be used. Wireless networks were established to generate a collection of training data. The dataset for the training phase was collected, examined, and processed before being used to build an ML model for prediction. The dataset is split into two subsets, "Input Series" and "Target Series", and then split into 70% training and 30% testing after it is loaded into the network as input. The input data should be normalized to fall within the range [0, 1], This corresponds to the actual highest and lowest values. A deep neural network learns over a series of iterations called epochs. In the first epoch, random initialization values are first assigned for the weight (w) and bias (b) parameters.

The input layer receives objects to observe data with known label values. These observations are usually categorized into packages (often called "mini-packages"). The neurons then perform their tasks and, if activated, pass the result to the next layer until the output layer provides a prediction. The prediction and the actual value are compared, and the difference between the two (which we call losses) is determined. The prediction with DL-LSTM networks is described in Algorithm 1.

Based on the results, revised weights and bias values are estimated to reduce losses, and these adjustments are propagated back to the neurons in the network layers. The next epoch repeats the preliminary stage of batch training with modified values of weight and displacement, which should increase the model's accuracy (by reducing losses). One thousand epochs is the maximum number that can be used for training. The learning rate is initially set at 0.1. The drop period for the learning rate is 125, and the drop rate is 0.2. We specify 0.2 for the layer dropout, which means that 20% of the layers are eliminated. We then add a dense layer, which changes the output to 1 unit. The model is then assembled using the well-known Adam optimizer, and the loss is measured by the mean-square error (MSE) loss function. In this way, the average value of the error squares can be determined. The model is then set up for 1000 epochs with a 32-batch size. The calculation can take several minutes depending on the technical data of the computer.

Algorithm1: Prediction using Deep Learning-LSTM network

```

1  InputSeries: Time series training data  $D = \{D_1, D_2 \dots, D_t\}$ 
2  TargetSeries: Time series training data  $t = \{t_1, t_2 \dots, t_n\}$ 
3  OutputSeries: the predicted output  $y = \{y_1, y_2 \dots, y_n\}$ 
4  Initialize: the weights and biases randomly
5  Split: data into 70% training and 30% testing data
6  size  $\leftarrow$  length(series) * 0.70
7  Train  $\leftarrow$  series [0 ... ize]
8  Test  $\leftarrow$  series [size ... length (size)]
9  Normalize: the dataset (Di) into values between 0 to 1
10 10  $\leftarrow$  nRepeatModel {e.g., Repeat each model 10 times}
11 1  $\leftarrow$  nDataTypes
12 1  $\leftarrow$  nDataCount
13 32  $\leftarrow$  batch size
14 1000  $\leftarrow$  number of epochs
15 Define: LSTM Network Architecture
16 while nDataCount  $\leq$  nDataTypes do
17     XTrain, YTrain, XTest, YTest, YPred  $\leftarrow$  loadInputData()
18     count  $\leftarrow$  1
19     for all nEpoch do
20         for all nBatch do
21             while count  $\leq$  nRepeatModel do
22                 define DL Model()
23                 compile Model()
24                 fit Model()
25                 train Model()
26                 evaluate Model()
27                 save model ()
28                 makePredictions()
29                 save Predicted Values()
30                 save AccuracyLoss()
31                 count  $\leftarrow$  count + 1
32             end while
33             count  $\leftarrow$  1 {Reset repeat counter}
34         end for
35     end for
36     nDataCount  $\leftarrow$  nDataCount + 1
37 end while

```

6. Simulation Results

This paper investigates how to predict V2X traffic using the DL-based LSTM model. We used the V2X network dataset (see Figure 1) for training the DL algorithm. We studied the prediction process in different situations according to the number of packets/s transmitted—4, 6, 8, 12, and 14 packets/s. We fitted the model to 1000 epochs with a batch size of 32 and an LSTM layer size of 200 hidden neurons. RMSE and MAPE were used to evaluate prediction accuracy. Table 1 displays the V2X traffic forecast accuracy using RMSE and MAPE.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (9)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{x_t} \right| \quad (10)$$

Table 2 shows the prediction accuracy of V2X traffic according to the numbers of packets transmitted of 4, 6, 8, 10, 12, and 14 packets per second in terms of RMSE, MAPE, and processing time.

Table 2. Comparison of prediction accuracy for V2X traffic using RMSE and MAPE.

Packets/s	MAPE	RMSE	Processing Time (s)
4	9.86	0.5427	78
6	10.77	0.6321	70
8	15.13	0.8175	71.3
10	20.76	1.1675	62.6
12	22.13	1.2622	56.8
14	28.33	1.4911	117.5

According to the above table, the best prediction accuracy was obtained at a transmission rate of 4 packets/s with an RMSE value of 0.5427 and a MAPE value of 9.86%. In this case, the maximum average improvement is 18.47%. With an RMSE value of 0.6321 and a MAPE value of 10.77%, the performance of 6 packets/s is almost identical to that of 4 packets/s. In this case, the maximum average improvement is 17.56%. For 8 packets/s, 10 packets/s, and 12 packets/s, the average improvement is 13.2%, 7.57%, and 6.2%, respectively. However, the models predicted at 14 packets per second have the worst prediction accuracy.

On the other hand, the models predicted at 12 packets per second have the fastest processing time, while the model predicted at 14 packets per second has the longest processing time.

Figures 3–8 illustrates the predicted throughput with a DL-LSTM model. Depending on the number of packets transmitted in the V2X network, we performed the procedures in different cases. Each figure has two curves, the first of which shows how the predicted traffic changes over time. The relationship between prediction loss and time is depicted by the second curve.

As can be seen from Figure 3, in the case of 4 packets transmitted per second, the prediction model that has the best prediction accuracy increases at 1 s, then gradually decreases until 11 s and becomes constant until 16 s, resulting in better accuracy. The second curve shows that the prediction loss is highest at 8 s and lowest at 1 s.

In Figure 4, using 6 packets transmitted per second, it can be seen that the predicted model increases at 1 s, gradually decreases until 6 s, and then remains constant until 16 s. The predicted model is the same as the predicted model. The second curve shows that the highest prediction loss occurs at 4 s and 8 s and gradually decreases until 15 s, and the lowest prediction loss occurs at 1 s and 6 s.

In Figure 5, with 8 packets transmitted per second, the prediction model increases at 1 s, then gradually decreases until 5 s, and then remains constant until 15 s. As can be seen in the second curve, in this case, the highest prediction loss occurs at 4 s and the lowest loss occurs at 1 s and 6 s.

Looking at Figure 6, using 10 packets transmitted per second, in the first curve the prediction model increases at 1 s, decreases to 6 s, remains constant to 8 s, and then increases slightly to 16 s. The prediction loss in the second curve is largest at 4 s and 8 s and gradually decreases until 15 s, and the smallest prediction loss occurs at 1 s and 6 s.

As can be seen in Figure 7, using 12 packets transmitted per second in the figure, the actual and predicted models increase over time. The predicted model increases at 1 s, decreases until 12 s, and then increases again until 16 s, as shown in the figure. The largest loss occurs between 6 and 10 s and then decreases until 15 s, while the smallest prediction loss occurs at 1 s, 4 s, 7 s, and 15 s.

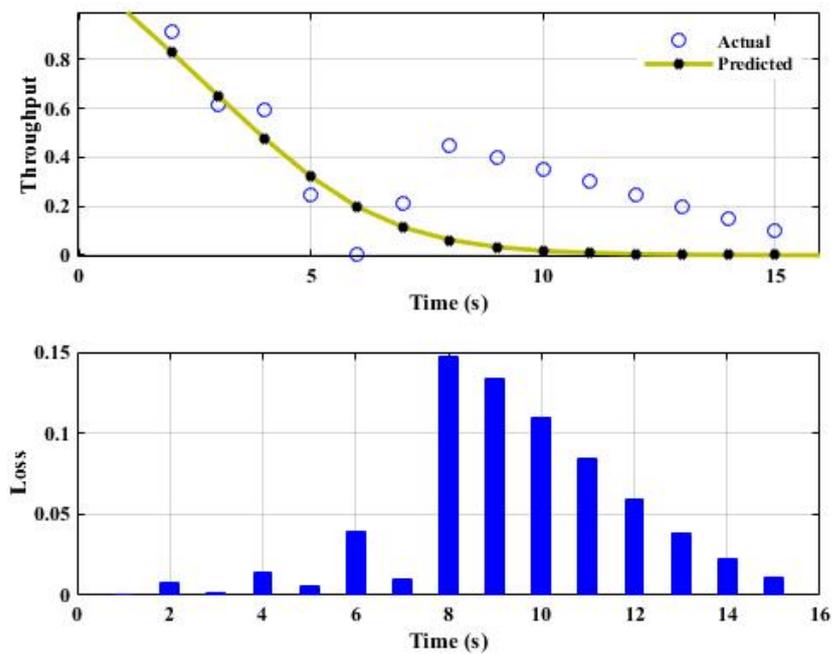


Figure 3. The predicted output at a transmission rate of 4 packets/s.

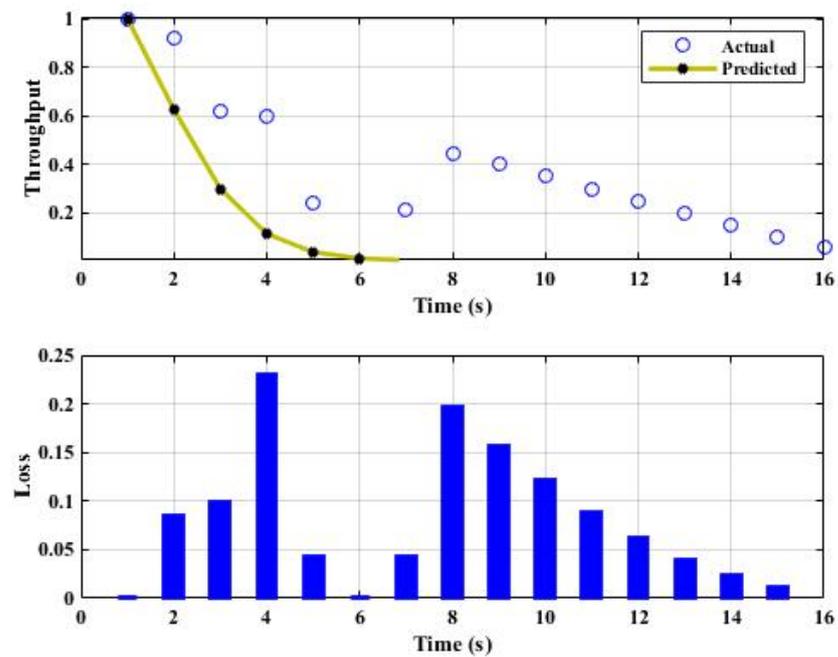


Figure 4. The predicted output at a transmission rate of 6 packets/s.

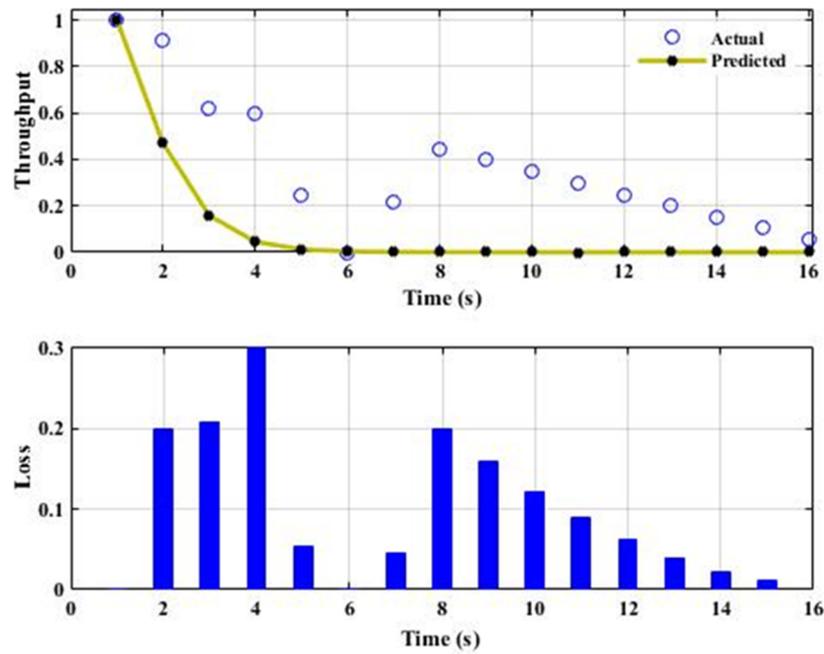


Figure 5. The predicted output at a transmission rate of 8 packets/s.

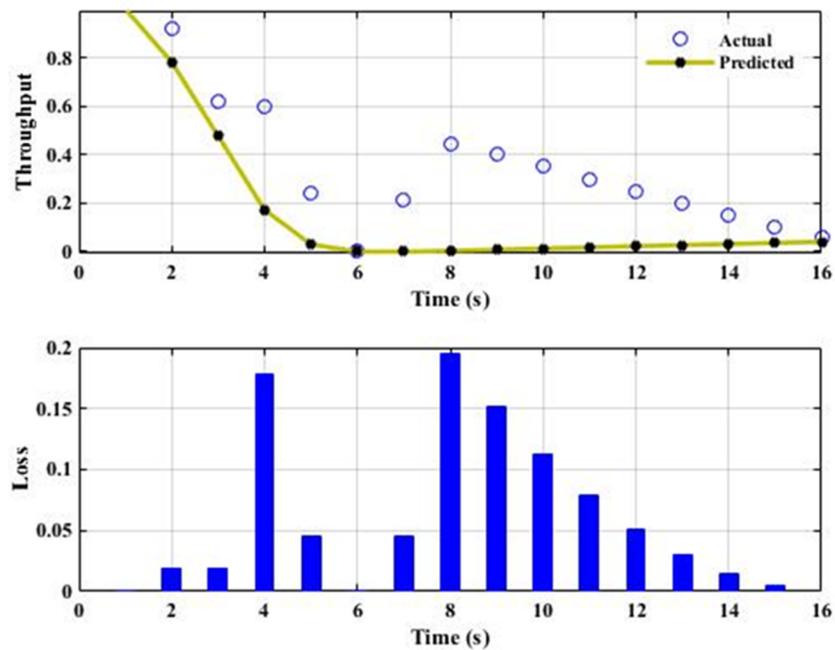


Figure 6. The predicted output at a transmission rate of 10 packets/s.

Figure 8 shows that throughput increases over time for both the actual and predicted models at a transmission rate of 14 packets per second. The predicted model increases at time 1 s, decreases until time 4 s, and then increases until time 16 s. While the lowest prediction loss occurs at time 1 s, it is highest at time 2 s to 4 s.

The trend is for the throughput rate to decrease over time where there is an obvious downward trend in some figures, as in Figures 3–5, and some figures that start with a downward trend and then turn into an upward trend, as in Figures 6–8. If we had a much longer series, we could see that these downward and upward trends are part of a long cycle.

These generated trends can go down or up depending on the observation data (DL training dataset) generated by the V2X network. If more packets are lost during transmission (one or more of these packets fails to reach their destination) due to network operation

issues, such as network congestion or security threats, throughput will decrease over time. However, if there is little or no packet loss, throughput will increase. Thus, based on the observed data, the trend for the predictive model is either up or down over time.

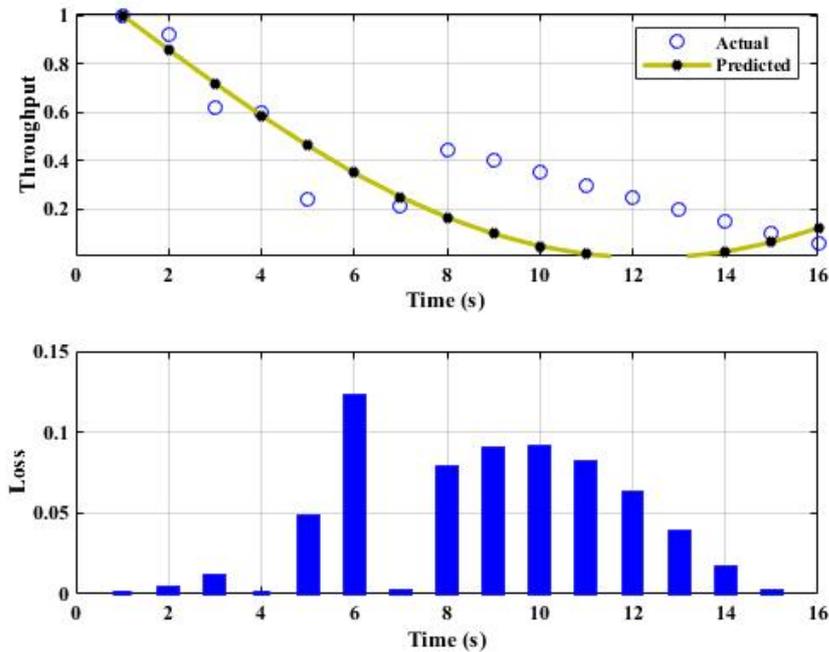


Figure 7. The predicted output at a transmission rate of 12 packets/s.

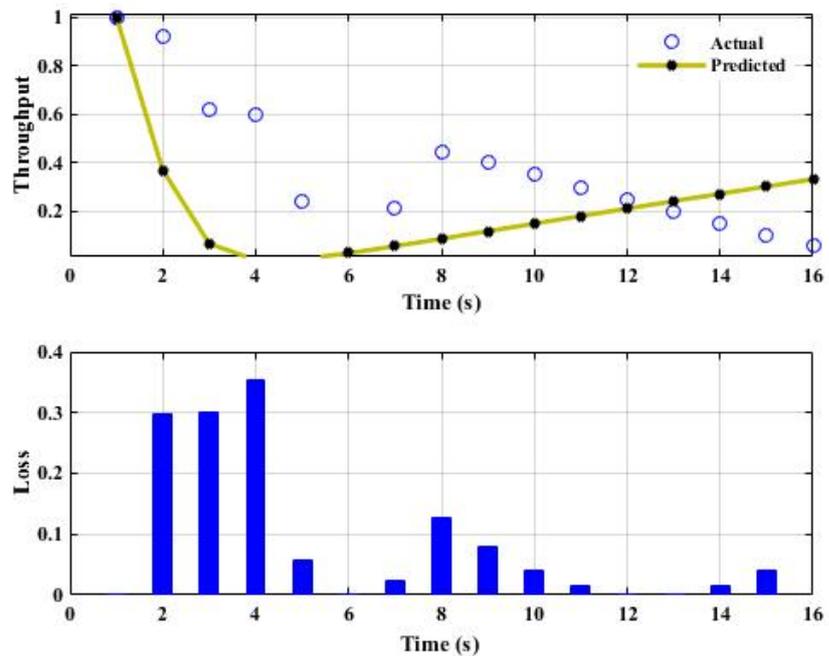


Figure 8. The predicted output at a transmission rate of 14 packets/s.

7. Conclusions

In this paper, the unidirectional LSTM with the DL model was proposed for V2X traffic forecasting. LSTM allows time series prediction models to predict future values as a function of past values. This allows for more accurate predictions, leading to better decision making. It also has the advantage of remembering values over a long time, producing a more accurate prediction of future values. According to the simulation results, the highest prediction accuracy was achieved with a number of transmitted packets of 4 packets/s,

which outperformed the competitors and represented outstanding performance. On the other hand, the models with a prediction of 14 packets/s have the lowest prediction accuracy compared to the others. The model with a prediction of 12 packets/s has the fastest processing time, but the models with a prediction of 14 packets/s have the slowest processing time compared to their competitors.

Author Contributions: Conceptualization, A.R.A. and A.K.; methodology, A.R.A.; software, A.R.A. validation, A.R.A., M.H.E., A.M., A.K. and A.R.A.; formal analysis A.R.A.; investigation, A.M. and M.H.E.; resources, A.K.; data curation, A.R.A. and A.K.; writing—original draft preparation, A.R.A.; writing—review and editing, A.R.A., A.K., M.H.E. and A.M.; visualization, A.R.A., A.M., M.H.E. and A.K.; supervision, A.K.; project administration, A.M. and A.K.; funding acquisition, A.M., A.K. All authors have read and agreed to the published version of the manuscript.

Funding: The studies at St. Petersburg State University of Telecommunications. M.A. Bonch-Bruevich were supported by the Ministry of Science and High Education of the Russian Federation by the grant 075-15-2022-1137.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The article contains the data, which are also available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sarker, I.H. AI-Based Modeling: Techniques, Applications and Research Issues towards Automation, Intelligent and Smart Systems. *SN Comput. Sci.* **2022**, *3*, 158. [[CrossRef](#)] [[PubMed](#)]
2. Salameh, A.I.; Tarhuni, M.E. Rom 5G to 6G—Challenges, Technologies, and Applications. *Future Internet* **2022**, *14*, 117. [[CrossRef](#)]
3. Abdellah, A.; Koucheryavy, A. *Survey on Artificial Intelligence Techniques in 5G Networks*; Telecom IT, SPbSUT: Saint Petersburg, Russia, 2020; Volume 8, pp. 1–10.
4. Monika, S. Integrating Artificial Intelligence and 5G in the Era of Next-Generation Computing. In Proceedings of the 2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST), Mohali, India, 17–18 December 2021; pp. 24–29.
5. Abdellah, A.R.; Alshahrani, A.; Muthanna, A.; Koucheryavy, A. Performance Estimation in V2X Networks Using Deep Learning-Based M-Estimator Loss Functions in the Presence of Outliers. *Symmetry* **2021**, *13*, 2207. [[CrossRef](#)]
6. Rekkas, V.P.; Sotiroudis, S.; Sarigiannidis, P.; Wan, S.; Karagiannidis, G.K.; Goudos, S.K. Machine Learning in Beyond 5G/6G Networks—State-of-the-Art and Future Trends. *Electronics* **2021**, *10*, 2786. [[CrossRef](#)]
7. Nassef, O.; Sun, W.; Purmehdi, H.; Tatipamula, M.; Mahmoodi, T. A survey: Distributed Machine Learning for 5G and beyond. *Comput. Netw.* **2022**, *207*, 108820. [[CrossRef](#)]
8. Abubakar, A.I.; Omeke, K.G.; Ozturk, M.; Hussain, S.; Imran, M.A. The Role of Artificial Intelligence Driven 5G Networks in COVID-19 Outbreak: Opportunities, Challenges, and Future Outlook. *Front. Comput. Netw.* **2020**, *1*, 575065. [[CrossRef](#)]
9. Tsourdinis, T.; Chatzistefanidis, I.; Makris, N.; Korakis, T. AI-driven Service-aware Real-time Slicing for beyond 5G Networks. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs), New York, NY, USA, 2–5 May 2022; pp. 1–6. [[CrossRef](#)]
10. Abdellah, A.R.; Koucheryavy, A. Artificial Intelligence Driven 5G and Beyond Networks. *Telecom IT* **2022**, *10*, 1–13.
11. Kaur, J.; Khan, M.A.; Iftikhar, M.; Imran, M.; Haq, Q.E.U. Machine Learning Techniques for 5G and Beyond. *IEEE Access* **2021**, *9*, 23472–23488. [[CrossRef](#)]
12. Gautam, K.K.S.; Kumar, R.; Sekhar, P.C.; Kumar, N.M.; Rao, K.S.; Chakravarthi, M.K. Machine Learning Algorithms for 5G and Internet-of-Thing (IoT) Networks. In Proceedings of the 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 21–22 April 2022; pp. 1–4. [[CrossRef](#)]
13. Gao, Z. 5G Traffic Prediction Based on Deep Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 3174530. [[CrossRef](#)]
14. Selvamanju, E.; Shalini, V.B. Machine Learning Based Mobile Data Traffic Prediction in 5G Cellular Networks. In Proceedings of the 2021 5th International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 2–4 December 2021; pp. 1318–1324. [[CrossRef](#)]
15. Trinh, H.D.; Giupponi, L.; Dini, P. Mobile Traffic Prediction from Raw Data Using LSTM Networks. In Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1827–1832. [[CrossRef](#)]

16. Chakraborty, P.; Corici, M.; Magedanz, T. System Failure Prediction within Software 5G Core Networks using Time Series Forecasting. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; pp. 1–7. [\[CrossRef\]](#)
17. Abdellah, A.R.; Alzaghir, A.; Koucheryavy, A. *Deep Learning Approach for Predicting Energy Consumption of Drones Based on MEC*; Koucheryavy, Y., Ed.; NEW2AN 2021/ruSMART 2021; Springer: Cham, Switzerland, 2022; Volume 13158, pp. 284–296. [\[CrossRef\]](#)
18. Zhu, Y.; Wang, S. Joint Traffic Prediction and Base Station Sleeping for Energy Saving in Cellular Networks. In Proceedings of the ICC 2021—IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021; pp. 1–6. [\[CrossRef\]](#)
19. Abdellah, A.R.; Volkov, A.; Muthanna, A.; Gallyamov, D.; Koucheryavy, A. Deep Learning for IoT Traffic Prediction Based on Edge Computing. In *Distributed Computer and Communication Networks: Control, Computation, Communications, DCCN 2020*; Vishnevskiy, V.M., Samouylov, K.E., Kozyrev, D.V., Eds.; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2020; Volume 1337, pp. 18–29. [\[CrossRef\]](#)
20. Wang, Z.; Hu, J.; Min, G.; Zhao, Z.; Chang, Z.; Wang, Z. Spatial-Temporal Cellular Traffic Prediction for 5 G and Beyond: A Graph Neural Networks-Based Approach. *IEEE Trans. Ind. Inform.* **2022**, *2022*, 1–10. [\[CrossRef\]](#)
21. Abdellah, A.R.; Koucheryavy, A. Deep Learning with Long Short-Term Memory for IoT Traffic Prediction. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*; Galinina, O., Andreev, S., Balandin, S., Koucheryavy, Y., Eds.; NEW2AN/ruSMART; Springer: Cham, Switzerland, 2020; Volume 12525, pp. 267–280. [\[CrossRef\]](#)
22. Abdellah, A.R.; Koucheryavy, A. VANET Traffic Prediction Using LSTM with Deep Neural Network Learning. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*; Galinina, O., Andreev, S., Balandin, S., Koucheryavy, Y., Eds.; NEW2AN/ruSMART; Springer: Cham, Switzerland, 2020; Volume 12525, pp. 281–294. [\[CrossRef\]](#)
23. Zhou, S.; Wei, C.; Song, C.; Pan, X.; Chang, W.; Yang, L. Short-Term Traffic Flow Prediction of the Smart City Using 5G Internet of Vehicles Based on Edge Computing. *IEEE Trans. Intell. Transp. Syst.* **2022**, *2022*, 1–10. [\[CrossRef\]](#)
24. Fitters, W.; Cuzzocrea, A.; Hassani, M. Enhancing LSTM Prediction of Vehicle Traffic Flow Data via Outlier Correlations. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 12–16 July 2021; pp. 210–217. [\[CrossRef\]](#)
25. Barmponakis, S.; Magoula, L.; Koursioupas, N.; Khalili, R.; Perdomo, J.M.; Manjunath, R.P. LSTM-based QoS prediction for 5G-enabled Connected and Automated Mobility applications. In Proceedings of the 2021 IEEE 4th 5G World Forum (5GWF), Madrid, Spain, 12–16 July 2021; pp. 436–440. [\[CrossRef\]](#)
26. Guerra-Gómez, R.; Boqué, S.R.; García-Lozano, M.; Bonafé, J.O. Machine-Learning based Traffic Forecasting for Resource Management in C-RAN. In Proceedings of the 2020 European Conference on Networks and Communications (EuCNC), Dubrovnik, Croatia, 15–18 June 2020; pp. 200–204. [\[CrossRef\]](#)
27. Chaalal, E.; Reynaud, L.; Senouci, S.M. Mobility Prediction for Aerial Base Stations for a Coverage Extension in 5G Networks. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin City, China, 28 June–2 July 2021; pp. 2163–2168. [\[CrossRef\]](#)
28. Tran, Q.H.; Fang, Y.-M.; Chou, T.-Y.; Hoang, T.-V.; Wang, C.-T.; Vu, V.T.; Ho, T.L.H.; Le, Q.; Chen, M.-H. Short-Term Traffic Speed Forecasting Model for a Parallel Multi-Lane Arterial Road Using GPS-Monitored Data Based on Deep Learning Approach. *Sustainability* **2022**, *14*, 6351. [\[CrossRef\]](#)
29. Alsaade, F.W.; Al-Adhaileh, M.H. Cellular Traffic Prediction Based on an Intelligent Model. *Mob. Inf. Syst.* **2021**, *2021*, 6050627. [\[CrossRef\]](#)
30. 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*, 42–49. [\[CrossRef\]](#)