Machine Learning with Variable Sampling Rate for Traffic Prediction in 6G MEC IoT

Jenifer.V ,Kg College of Arts and Science,Coimabtore

**Abstract**

The high-speed development of mobile broadband networks and IoT applications has brought about massive data transmission and data processing, and severe traffic congestion has adversely afected the fast-growing networks and industries. To better allocate network resources and ensure the smooth operation of communications, predicting network traffic becomes an important tool. We investigate in detail the impact of variable sampling rate on traffic prediction and propose a high-speed traffic prediction method using machine learning and recurrent neural networks. We first investigate a VSR-NLMS adaptive prediction method to perform time series prediction dataset transformation. Then, we propose a VSR-LSTM algorithm for real-time prediction of network traffic. we simulate the prediction accuracy of the VSR-LSTM algorithm based on the variable sampling rate proposed. The experiment shows that VSR-LSTM has higher traffic prediction accuracy because its sampling rate varies with the traffic.

# Introduction

As the global mobile industry moves toward 6G networks, mobile edge computing (MEC)-based network infrastruc- ture has received unprecedented attention to support In- ternet of Things (IoTs) applications with diverse business needs [[1].](#_bookmark17) To better serve users, the connectivity and in- telligence provided by edge computing have tremendous advantages in terms of real-time services, smart living, se- curity, and reliability [[2].](#_bookmark18) At present, edge computing has been applied in smart campuses, video surveillance, in- dustrial IoTs, augmented reality/virtual reality (AR/VR) and other application scenarios, which also proves that MEC- based network infrastructure is efective and fully capable. The capabilities of edge computing rely on edge servers, which are usually deployed with base stations. It is expected that more IoTs applications will be carried out based on MEC in the future, and the massive data they generate will have great demands on network resources such as band- width, computing power, and storage [[3,](#_bookmark19) [4].](#_bookmark20) Therefore, in

future, multiple MEC servers will be needed to jointly provide services for diferent applications. Since MEC servers have diferent computing processing capabilities and are deployed in a distributed manner, it is critical to offioad computing tasks to these heterogeneous MEC servers according to diferent application requirements. Therefore, reasonable and efective resource allocation mechanisms according to the usage of wireless network resources and MEC server resources will efectively ensure the service requirements of users.

Unreasonable resource allocation will not only increase the operating costs of communication networks but also may lead to serious energy waste. Accurate wireless network traffic prediction can intuitively reflect the changing trend of service requirements, provide an important refer- ence for communication network resource allocation, and is an important guarantee to achieve reasonable and efficient resource allocation and computing task offioading [[5,](#_bookmark21) [6].](#_bookmark22)

It has always been a meaningful research topic to analyze the distribution and demand of communication traffic by predicting the wireless network traffic to guide the com- munication resources’ allocation [[7].](#_bookmark23) The previous wireless traffic prediction adopted manual prediction and statistical- based prediction methods, and their limitations are obvious. Manual prediction is inefficient and cannot adjust the network resource allocation in real-time according to the change of service demands. The statistics-based traffic prediction models intelligently utilize certain statistical characteristics, but cannot efectively comprehensively uti- lize various information that has an important impact on wireless traffic [[8].](#_bookmark24) As machine learning techniques are intensively researched, there have been some works using machine learning and deep learning algorithms for wireless network traffic prediction [[9–](#_bookmark25)[18].](#_bookmark34)

In the future, artificial intelligence will be one of the native

features of the next-generation mobile communication net- work, namely the sixth-generation (6G). Through AI-based endogenous intelligent design in air interface algorithms, wireless network architecture, and wireless traffic prediction, etc., 6G-based communication network can better achieve network autonomy and intelligence, thus realizing intelligent operation and maintenance management of 6G network in- frastructure, including MEC servers and efficient automatic deployment of services [[19,](#_bookmark35) [20].](#_bookmark36) The traffic prediction model based on AI algorithm can automatically mine various fea- tures contained in wireless data and comprehensively use these features to accurately predict wireless traffic in real-time. Accurate traffic prediction can directly reflect the spatial and temporal distribution of communication service demands, guide the network resources allocation in diferent network nodes, and then offioad thecomputing tasks in diferent MEC servers, thereby improving user service experience and en- hancing the autonomous and intelligent operation and maintenance of communication networks.

Machine-to-machine (M2M) communications, auton-

omous driving, and virtual reality are just a few of the new applications that 6G cellular networks are expected to make possible in the next few years. These applications all call for better network latency, capacity, and context awareness. It is critical to make the network aware of traffic demands to achieve these strict standards. The development of an in- telligent network necessitates traffic analysis and accurate forecasting of user demand. Knowing user demand ahead of time allows the network to allocate resources more effi- ciently. The network can manage resource distribution between users who are competing for resources promptly. The remainder of this paper is organized as follows.

Related works closely to our research are listed in Section [2.](#_bookmark0) A traffic prediction model for 6G MEC IoT is described in Section [3.](#_bookmark1) Section [4](#_bookmark6) focuses on traffic prediction methods with variable sampling rates(VSR) using machine learning and RNN techniques. In this section, the VSR-NLMS adaptive prediction method to perform time series predic- tion dataset transformation and the VSR-LSTM algorithm for real-time prediction of network traffic are proposed. Simulation and performance analysis are presented in Section [5.](#_bookmark13) Section [6](#_bookmark16) concludes the full text.

# Related Works

Network traffic analysis and prediction are fundamental to traffic engineering, network planning, optimization, ad- ministration and maintenance, resource allocation, load balancing, etc.. The previous wireless traffic prediction adopted manual prediction and statistical-based prediction methods, which have lots of flaws such as Low efficiency and non-real-time. As machine learning techniques are inten- sively researched, there have been some works using ma- chine learning and deep learning algorithms for wireless network traffic prediction.

All of [[9–](#_bookmark25)[14]](#_bookmark30) address network traffic prediction problems in the telecom network. In these research works except [[12],](#_bookmark28) LSTM or LSTM variants has been adopted and achieved good prediction performance compared with other algorithms such as ARIMA, SVR, FFNN, RFR, KNNR, etc.In [[9–](#_bookmark25)[11,](#_bookmark26) [14],](#_bookmark30)

the prediction focused on aggregated behavior, e.g., con- sidering traffic volumes observed over a given time interval (normally 5–15 minutes), which is coarse-prediction.

In [[12,](#_bookmark28) [13],](#_bookmark29) the author investigates and specializes a set of architectures selected among convolutional, recurrent, and composite neural networks to predict mobile-app traffic at the finest (packet-level/mobile app) granularity. To provide the AI-native services for the 6G vision, the author in [[14]](#_bookmark30) proposed a novel edge-native framework to provide an intelligent prognosis model using LSTM-based encoder- decoder for data traffic prediction. The prognosis model was trained on real time-series multivariate data records col- lected from the edge *μ*-boxes of a selected testbed network. In [[15–](#_bookmark31)[18],](#_bookmark34) the traffic predictions of Internet or campus network, such as ARQ message and ping command, are all fine-grained prediction. Especially in [[18],](#_bookmark34) wavelet transform is used to preprocess the data before prediction, trans- forming the 1-dimensional time series data into 3-dimen- sional data, which is more conducive to GRU (a LSTM variant) feature extraction and then obtained a better per-

formance than RNN.

A detailed comparison of these related works has been compared according to the aspects of research objects, traffic data granularity, used data sets, and algorithms in this paper, as shown in Table [1.](#_bookmark2)

LSTM has better performance in predicting time series data such as network traffic. But al- though not specified in the literature, especially telecommu- nications network traffic prediction, the traffic data generated using a fixed sampling rate, without considering traffic speed changes, will lead to large complexity of computation or predict the problem of inaccuracy. Aiming at the above problems, a new algorithm, VSR-LSTM, which combines variable sampling rate and LSTM, is proposed in this paper.

In addition, for readers’ convenience, all the acronyms in this paper have been collected and listed in Table [2.](#_bookmark3)

# System Model

As shown in Figure [1,](#_bookmark4) the mobile edge computing system model for machine learning-based traffic prediction is a

Table 1: Related works performing prediction of diferent research objects and by means of diferent techniques.

[Ref] Research object Fine or coarse Dataset Simulator Techniques

ARIMA

[[9]](#_bookmark25) Telcom network Coarse

(15 min)

GEANTWIDE Not mentioned

SVR LSTM RCLSTM

SR-based ARIMA

Telcom user traffic and location

[[10]](#_bookmark27)

Coarse (15 min)

GENAT Not mentioned

SVR LSTM RCLSTM FFNN ARIMA

1. Telcom. Network Coarse Operator data Python

FFNN LSTM

1. Mobile APP Fine MIRAGE-2019 Not mentioned HMM

RFR

LR K-NNR

RFR

1. Mobile APP Fine MIRAGE-2019 Not mentioned
2. Mobile 6G network Coarse (5 min) Locally obtained Edge *μ*-boxes, jupyter

notebook

MC CNN LSTM GRU

LSTM-based encoder and decoder

University campus datacenter

[[15]](#_bookmark31)

Fine EDU1 dataset Python, keras CNN DNN

1. SDN controller (ONOS) Fine Ping ARQ message Not mentioned

RF

1. Internet Fine DNS traffic Python, TensorFlow

SF LDA SVR BPNN LSTM

1. Internet Fine User data MATLAB GRU

RNN

Table 2: Acronyms list in this paper.

three-tier hierarchy consisting of a cloud platform, multiple

MEC gateways, and a large number of end-users, including

Acronyms Full name

ARIMA Auto regressive integrated moving average ARQ Automatic repeat response

CNN Convolutional neural network DNN Deep neural network

FFNN Feed forward neural network FSR Fixed sampling rate

GRU Gated recurrent unit

HMM Hidden markov models

MEC Mobile edge computing NRMSE Normalized root mean square error KNNR K-nearest neighbor regressor

LSTM Long short-term memory

LDA Linear discriminant analysis LR Liner regression

MC Markov chain

RCLSTM Random connectivity LSTM RF Random forest

RFR Random forest regressor

RNN Recurrent neural network

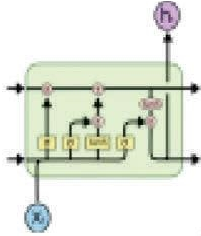
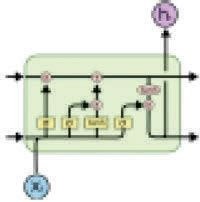
SR-based Sparse representation-based

SVR Support vector regression

VSR Variable sampling rate

multiple independent IoT networks. Each IoT network serves many end-users (i.e., diferent end devices). Diferent terminal types and usage scenarios have diferent comput- ing, storage, and communication capabilities. For example, smartphones have relatively high computational storage and communication capabilities, and rechargeable batteries have high energy supply capabilities. But some IoT nodes have deficient capacity compared to smartphones, especially many nodes that cannot replace batteries, and their com- putational storage and communication capabilities are greatly limited. At the same time, various services have diferent QoS requirements for latency, energy consump- tion, communication bandwidth, and other indicators, which require diferent processing methods for diferent terminal types and diferent business needs. Generally speaking, services with low computation and power re- quirements but high latency requirements can be executed on local terminals with strong capabilities, such as smart- phones. For IoTnodes with limited computation and power, data can only be transmitted to gateway nodes or edge servers for processing, and for that kind of computationally

Cloud Server



...

Edge Server & Gateway

EnodeB EnodeB

Edge Server & Gateway

...

Figure 1: Traffic prediction for 6G MEC IoT.

intensive service, they also need to be offioaded to remote servers with unlimited energy computing power to process them.

In Figure [1,](#_bookmark4) whether IoT end-users offioad tasks to edge servers for execution or edge servers offioad computationally intensive tasks to remote ends for execution, both require the system to allocate appropriate network bandwidth re- sources. Although 6G has increased the network speed and bandwidth a lot compared to 5G, the competition for network resources, especially bandwidth resources, still exists, and this resource competition may become more intense in the future, requiring dynamic control of network resources based on user demand. Traffic prediction is of great significance to achieving dynamic resource allocation and is a prerequisite and guarantee for the edge computing server to achieve dynamic resource allocation.

In addition to being able to complete the execution of tasks offioaded to it by users and offioad tasks with greater

discovered that the main source of traffic load volatility is the timeliness features of mobile user behavior. A notable ex- ample of this is the fact that students’ real-time traffic on campus is scheduled and determined by the calendar. Then, the real-time network traffic load is also limited by the number of visits and applications. When the number of users fluctuates quickly, the fast-changing zone appears, but when the number of users remains consistent, the slow- changing zone appears. Based on these findings, the best VSR strategy is to sample at a low sampling rate in slowly changing regions and at a high sampling rate in rapidly changing regions. The constant sampling rate approach is often used when considering the entire traffic load distri- bution, whereas the variable sampling rate strategy is used when the traffic load distribution needs to be sampled and reconstructed independently, a practical scenario where the traffic load varies in speed from region to region. The maxi- mum frequencies in the slow-change and fast-change zones,

computational demand to the cloud for execution, the edge

max

max

respectively, are represented by *fs* and *ff* . From Nyquist’s

server in Figure [1](#_bookmark4) should also have dynamic traffic prediction

theory, the sampling rate must be twice as fast as *fs* in the

functions to accurately predict the bandwidth resources

slow-changing region and twice as fast as *ff*

max

required for various offioading tasks and provide a basis for

*VSR*

max in the fast- changing region. Therefore, the average sampling rate *R*avg :

network bandwidth resource allocation. The cloud server

*f*

*R*avg

�

max *s* max *f ,* (1)

to it by the edge servers.

*T*

mainly completes the computation-intensive tasks offioaded

2*fs T* + 2*f T*

*VSR*

*s*

+ *T*

*f*

# Prediction Methods

of which *Ts* and *Tf* represent the total length of time covered by the slow and fast-change zones, respectively. Considering

* 1. *The Significance and Preliminary Solution of Developing Variable Sampling Rate.* To avoid confusion in the forecast, if a constant sample rate is to be utilized, it must be double

we have

*s*

max

*f*

*f*

max

< *f*

� *f*max*,* (2)

the maximum frequency of the traffic load profile. Since more traffic sample operations likewise generate more traffic

avg

*VSR*

*R*

< 2*f*

max

(3)

prediction operations, a constant sampling rate puts sig- nificant computing complexity into the system. As a result, developing a low-sampling-rate solution seems appealing.

The traffic load curve, which has areas with slow and rapid changing movements, served as inspiration for the notion of VSR (variable sampling rate) [21, [22].](#_bookmark38) We

indicating that the sampling rate of the VSR method is lower than that of the constant sampling rate method.

* 1. *VSR-NLMS Adaptive Forecasting Method.* As the load profile is unknown at the time of prediction, it is not possible

to classify the load profile into low- and high-speed types. The author in [21] created the VSR-NLMS adaptive fore- casting method, which combines the FSR-NLMS (fixed step size-NLSM) predictor with VSR. The sampling rate of time *tn* is defined as

1

*4.4. RNN and LSTM.* The earliest known RNN is Hopfield Network (HN) proposed by Hopfield in 1982 [[24].](#_bookmark40) RNN, as a neural network model that can process time series and structural series data, has been widely used in handwriting recognition, speech discrimination, text translation, and other fields. The data contained in these fields have a

*Rs*(*n*) � Δ*t*

*n*−1*,n*

*.* (4)

common feature. That is to say, the input samples are all continuous sequence data, which can be a piece of text or a

in the VSR-NLMS scheme, and it is iteratively updated to show a negative correlation with the target prediction error bound *Eb* > 0. We further have the following constraint

*R*min ≤ *R* (*n* + 1) ≤ *R*max*,* (5)

*s*

*s*

*s*

where, *R*max and *R*min denote the system’s highest and lowest

piece of speech. There is a great correlation between the preceding and the following, and the length of the data is diferent, which cannot be accurately divided into separate training samples by traditional neural networks. Compared

connect the output at the current moment with the output at

with the traditional neural network, the RNN model can

*s s* the previous moment so that the neural network has the

sampling rates, respectively.

As the flow load curve is changing rapidly resulting in large errors, the VSR-NLMS scheme is used here to update the sampling rate in time to ensure accuracy. Based on the sub- sequent observation, the VSR-NLMS method adjusts the sampling rate. To achieve forecast accuracy, the sampling rate must be raised because the greater the forecast error, the greater the complexity of the traffic load situation. A small prediction error, on the other hand, indicates that the traffic load profile varies slowly, allowing the sampling rate to be adjusted to reduce computing complexity. When deciding on *R*max, we must choose between prediction accuracy and compilation efficiency. Large *R*max values, obviously, can result in reduced prediction errors, but they also increase the computational complexity of the sampling and prediction method. When we choose *R*min, we infer that the linear predictor’s reaction time cannot be too long to handle bursty traffic.

*s*

*s*

*s*

*4.3. Dimensional Transformation in Time-Series Prediction.* Network traffic prediction is a temporal sequential fore- casting technique, the core idea of which is to analyze the nonlinear correlation between the previous data and its historical data at a certain time point [[23].](#_bookmark39) The prediction of values at a future point in time is accomplished based on the outcomes of the modeling analysis. We must use traffic flow time series modeling in this paper. Traffic flow data was converted into multidimensional data, including input feature vectors and model output sample labels in a format to better model LSTM. Scholars now use the “sliding win- dow” approach to convert one-dimensional (1D) data into two-dimensional (2D) data. The following are the primary phases in this method for properly converting 1D time series into 2D machine learning data type.

Step 1: Choose moment *T*and collect *N* historical values before moment *T* and set them as feature vectors. *N* is the length of the feature vector.

Step 2: Construct the output vector by taking moments

*T +* 1 to *T + M* as the label values. *M* is the number of

output variables, which represents the step size of the prediction.

function of “memory”. It can record the historical sequence information, continuously reduce the gradient error be- tween the predicted value and the real value in the process of iteration, and finally obtain the optimal model. Normally, any sequence can be obtained by predicting through the RNN model.

However, in the actual training process of the RNN model, it is found that it is still a little insufficient to store the amount of historical information. Since the RNN model stores the corresponding historical information by the number of network layers, the less the number of network layers, the more incomplete the historical information recorded. In addition, the more the number of network layers, the more complex the training process will be, and it is easy to have the phenomenon that the gradient descent speed is fast or even disappears, both of which will lead to poor prediction performance of the model [[25].](#_bookmark41)

To solve the above problems, the long short-term memory network (LSTM) model is proposed by Hochreiter et al. in 1997 [[26].](#_bookmark42) Subsequently, the LSTM model quickly made great achievements in speech recognition and machine translation. LSTM, as a variant of RNN, can solve the common problems in the RNN model by changing the structure of the hidden layer. Since the LSTM model is obtained by changing the RNN model, the two models have the same output layer and input layer structure, and the diferences are mainly reflected in the structure of the hidden layer.

The structure of the hidden layer of the LSTM model is shown in Figure [3.](#_bookmark9)  The input gating unit *It* can be used to control, where information can be saved to the memory unit at the current moment, and it consists of two network layers with acti- vation functions of sigmoid function *σ* and tan h function, respectively.

*It* � *σ*( *Aimt*−1 + *Bi𝑥t* + *bi,*

� tan *h*( *Acm*

+ *b ,* (6)

*C*

*t*

*t*−1

+ *Bc𝑥*

*t*

*c*

The basic flow chart of the “sliding window” based method is shown in Figure [2.](#_bookmark8)

where *xt* denotes the input vector at moment *t*, *mt* − 1 is the output of the previously hidden layer neuron node, and by

One dimensional time series (network traffic flow)

...

...

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| XT-3 | XT-2 | XT-1 | XT | XT+1 | XT+2 | XT+3 | XT+4 | XT+5 |



Sliding window (M+N)

XT+M

|  |  |  |  |
| --- | --- | --- | --- |
| XT-1 | XT | XT+1 | XT+2 |

...

XT-N-1

...

...

...

Feature vector (M) Sample label (N) Sliding window (M+N)

XT+M+1

XT-N

vector

Feature

|  |  |  |  |
| --- | --- | --- | --- |
| XT | XT+1 | XT+2 | XT+3 |

...

...

...

...

Feature vector (M) Sample label (N)

Sample

label

...

XT-N-1

...

XT-N

Two dimensional time series (network traffic flow)

...

...

XT+M+1

XT+M

|  |  |  |  |
| --- | --- | --- | --- |
| XT-1 | XT | XT+1 | XT+2 |

|  |  |  |  |
| --- | --- | --- | --- |
| XT | XT+1 | XT+2 | XT+3 |



Training model

Figure 2: Time series prediction data set transformation.



σ

σ

Input Gate

*I*t

Output Gate *O*t

[x , m ]

Memory Unit

t t-1

Tash

*Ct*

X

*C*

t

Tash

X

m

t

σ

Forget Gate

*F*t

X

Figure 3: LSTM model hidden layer structure.

using the sigmoid function to activate *𝑥t* and *mt* 1. *Ai* de- notes the weight value corresponding to the output result of the previously hidden layer neuron at the input gate, while *Bi* is the weight value of input vector *𝑥t* and *bi* is the bias

−

parameter of the input gating unit at the time of calculation.

*Ct* the candidate cell state, *Ac*, *Bc*, and *bc* are weight values and the bias parameter of *Ct* respectively.

Next, the result of multiplying the elements corre- sponding to the two results is used for the update of the memory block cell.

The forgetting gating unit *Ft* is mainly used to link the state of the memory block unit at the previous moment with the state of the memory block unit at the current moment.

The final result obtained through the forgetting gate

*Ft* � *σ Afmt*−1 + *Bf𝑥t* + *bf ,* (7)

where *Af* denotes the weight value corresponding to the output result of the previously hidden layer neuron at the

Then, we need to go back in time and use the error of the latter time state to calculate the error of the previous time state.

At state *tn*, the error of the input gating unit is

*t*

input forgetting gate. The weight of the information entering the forgotten gating unit through the input gating unit at the

*ϑi* � *ϑt* ∘ *Ot* ∘ *Ct* ∘ *it* ∘ ( 1 − *it* ∘ 1 − tan *h*( *ct*2*.* (11)

current moment is denoted by *Bf*, and *bf* is the bias pa-

The forgetting gating unit error is

rameter of the forgotten gating unit at the time of

calculation.

The output gating unit is composed of two parts, the

*ϑ* � *ϑ* ∘ *o* ∘ *c* ∘ *f* ∘ ( 1 − *f*  ∘ 1 − tan *h*( *c* 2*.* (12)

current moment input vector combined with the informa-

tion obtained from short-term memory output results *Ot*

and the output result of the information obtained by

combining the input vector with long-term memory at the current moment *mt*. The activation functions used are the sigmoid function and the tan *h* function, respectively.

( −

The output gating unit error is

*ϑot* � *ϑt* ∘ tan *h ct* ∘ *t* ∘ 1 − *ot .* (13)

( (

*ft*

*t*

*t*

*t*−1

*t*

*t*

*t*

The error of the memory block unit is

*ϑc* � *ϑt* ∘ *ot* ∘ *it* ∘ ( 1 − *c*2 ∘ 1 − tan *h*( *ct*2*.* (14)

*t*

*t*

*Ot* � *σ A*∘*mt* 1 + *B*∘*𝑥t* + *b*∘ *, mt* � *Ot* ∘ tan *h*( *Ct,*

(8)

Using the above four values, the error value of the state

*tn*−1 can be calculated as

where, ∘ is the element-wise multiplication, which imple-

ments the product of the corresponding positional elements

of two matrices. As mentioned above, *A*∘, *B*∘, and *b*∘ are weight values and the bias parameter of the *Ot* respectively. The state of the memory block unit is determined by both

the past moment state and the current moment state, where the past moment state is obtained by multiplying the past moment unit state with the output result of the forgetting gating unit in accordance with the corresponding element, and the current moment state is obtained by multiplying the current moment unit state with the current moment input gating unit in accordance with the corresponding element.

*Ct* � *Ct*−1 ∘ *Ft* + *It* ∘ *Ct* (9)

After the above discussion, we have clearly understood the importance of controlling the sampling rate. The core of this article is to continuously adjust the sampling rate by analyzing the error. The processing method of this article is to apply the LSTM algorithm. How does the LSTM algo- rithm work in the past? The error value is input, forgotten, and output as a sample, which will be discussed below.

The reason why the LSTM algorithm is diferent from the traditional RNN algorithm is that it can delete some un-

necessary data through the forget gate. Among them, *A* , *B* ,

*ϑt*−1 � *Aoϑot* + *Afϑft* + *Aiϑit* + *Acϑct.* (15)

Then, the gradient of the bias term and the weight gradient *A*, *B*, and *b* are derived by the chain rule, and then the gradient is updated by the gradient descent method. To be accurate enough for the calculated *A*, *B*, and *b* values, after training with a sufficient time-span, we use the sampling error of the previous time state *Eb* as the object to use the

LSTM algorithm to predict. When *Eb* is too high, it means

that the sampling accuracy is not enough and we need to

increase the sampling rate; when the value *Eb* is too low, it means that the sampling rate is too high and too many computing resources are wasted. In this case, the sampling rate needs to be reduced.

In addition, it should be pointed out that gated recurrent unit (GRU) is also a variant of RNN, and its structure is similar to LSTM, with one less gate than LSTM, which reduces matrix multiplication and can save a lot of time without sacrificing performance in small dataset scenario. But in the scenario of large datasets, LSTM has better performance than GRU, such as prediction accuracy, recall, AUC, etc. In our study, the dataset obtained from the MEC servers is large, so LSTM with better performance is used for prediction in this paper [[27].](#_bookmark43)

*i i*

and *bi* in the input gating unit; *Af*, *Bf*, and *bf* in the forgetting gating unit; and *Ao*, *Bo*, and *bo* in the output gating unit need to be obtained by the LSTM model through training. The training method is as follows: the error at time *t* is defined in LSTM as *ϑt* and the sum of squares of all node errors in the output layer is represented by *L*. At this time, we have the following formula

* 1. *The Proposed VSR-LSTM Model.* In this paper, we make full use of the excellent advantage of LSTM in time series prediction, which combines the idea of variable sampling rate, improves the VSR-NLMS algorithm, proposes a net- work highway traffic prediction model based on LSTM variable sampling rate, named as VSR-LSTM, which con-

1 *τ* tains the following three main parts.

*L* � 2 ( *yt* − *yt,*

*t*�1

(10)

* + 1. Transform the original data form and divide the

*zL*

*ϑt* � *zm .*

*t*

Among them, *yt* is the prediction value, *yt* is the true value, and *mt* represents the hidden state.

training set and test set.

* + 1. Train the LSTM neural network.
    2. Realize traffic flow prediction and verify the results. The model’s fundamental stages are listed below.

Table 3: Summary of notations in Section [3.](#_bookmark1)

Symbol Description

The total length of time covered by the fast-change zones

*Tf*

The total length of time covered by the slow-change zones

*Ts*

Table 4: Training hyperparameters.

Hyperparameters name Value

Initial learning rate 0.001

Number of epochs 100

LSTM hidden states 64

LSTM hidden layers 5

*s*

*f*

m*f* ax

Maximum frequencies in the slow-change

Optimization algorithm Adam

*f*max Maximum frequencies in the fast-change Loss function MAE

*R*

avg

*VSR*

Average sampling rate

Rs(n) The sampling rate of time *tn*

*E b* Target prediction error bound

determine the aggregate cell traffic for each server. We use

*s* max *s* min

*R*

*R*

������������

*Ot* The output of the model at instant *t*

*N*

2

The system’s highest sampling rates The system’s lowest sampling rates

normalized root mean square error (NRMSE) as a measure of the prediction algorithm’s efficacy, which is defined as

*It* Input gating unit at instant *t*

*Ft* Forgetting gating unit at instant *t C* The internal state

NRMSE 1

*𝑥*

�

*t*�1 ( *t* − *𝑥t* *,* (16)

*mt* The hidden state

*N*

*𝑥t* Input vector at moment

*ϑt* The error at time *t* is defined in LSTM

*A c*, *Af*,*Ao* The weight values of *xt*

*B c*, *Bf*, *Bo* The weight values of *mt* − 1

*b c*, *bf*, *bo* The bias parameters

The sum of squares of all node errors in the output

*L*

layer

Step 1: Based on the actual collected traffic flow data, the training set and test set are divided in the ratio of 7 : 3.

Step 2: Based on the basic principle of “sliding window” method, the original one-dimensional traffic data is converted into two-dimensional data.

Step 3: Preprocess the raw data by normalization and other methods.

Step 4: Input the training data into the LSTM network, train the model parameters, and build the prediction model.

Step 5: Input the input vectors of the test set into the trained LSTM neural network and compare the pre- diction results with the real values to get the error of the model.

The summary of notations in Section [3](#_bookmark1) is listed in Table [3.](#_bookmark11)

# Numerical Results

In this section, we implemented the proposed VSR-LSTM and compared it with the traditional fixed sampling rate algorithm FSR-NLSM. Tto facilitate performance compar- ison, the original fixed sampling step size algorithm FSR- NLMS is also implemented based on LSTM, namely FSR- LSTM.

* 1. *Evaluation Setup.* We assess the performance of the suggested architecture using a collection of mobile traffic statistics from ten separate MEC servers that we gathered over the course of 1 month. According to Section [4.5,](#_bookmark10) we

where *𝑥t* and *xt* are the predicted value and its corresponding observation at the time *t*, respectively, and *𝑥t* is their mean. *N* is the total number of points. The accuracy of the suggested architecture is compared using the same metric with that found using other prediction algorithms.

Table [4](#_bookmark12) reports the selected hyperparameters. One of the hyperparameters that must be chosen that might influence the trade-of between the amount of time and the prediction accuracy required to train the network is the number of hidden layers, which is fixed at 5. The amount of information that must be main- tained and utilized by the network is determined by the connection between the quantity of earlier observed values and the accuracy of the multistep prediction, which is the focus of our attention. The prediction accuracy could be enhanced by adding more layers. We set the total number of epochs to 100 for the same reason. The architecture was trained and validated using three weeks of data. The fol- lowing results relate to the previous week. We iteratively change the network weights based on the training data using the Adam optimization.

* 1. *Results’ Analysis.* The results of multistep prediction, which involves making predictions for future time instants while delaying the output by a predefined number of timeslots, are then shown. We demonstrate how accuracy sufers when we attempt to forecast traffic statistics for upcoming timesteps. We also look at how the length of the timeslots *T* and the number of observations that the LSTM network can observe afect the results. These design pa- rameters must be computed since they have an impact on the LSTM network’s memory capacity and the amount of traffic data that must be kept for an accurate prediction.

Finally, we compare the proposed algorithm, i.e., VSR- LSTM with a classical time-series network traffic prediction method (FSR-LSTM). For a fair comparison, the same number of hidden layers are used.Figure [4](#_bookmark14) illustrates the traffic prediction for the same time-span using both methodologies. Using the FSR-NLMS model has lower ac- curacy because the predictions tend to be closer to the mean of the flow, while VSR-LSTM has a higher flow prediction

4 4

3 3

Traffic [Gbit/s]

Traffic [Gbit/s]

2 2

1

0

Ground Truth FSR-LSTM

TIME

(a)

1

0

Ground Truth VSR-LSTM

TIME

(b)

Figure 4: Traffic prediction carve obtained with diferent models. (a) FSR-LSTM. (b) VSR-LSTM.

0.3

0.2

NRMSE

0.1

0 1 2 3 4 5 6 7 8 9 10

MEC server

VSR-LSTM FSR-LSTM

Figure 5: Traffic prediction errors obtained with diferent models.

accuracy because its sampling speed increases with changes in flow. Furthermore, for the two prediction methods on the 10 flow profiles, we compare their mean errors. As expected, the proposed algorithm obtains less prediction error of the moving flow due to the VSR properties and relative to the FSR-LSTM model, as shown in Figure [5.](#_bookmark15)

The diference in computational complexity between the proposed VSM-LSTM and the baseline FSR-LSTM is mainly due to the diference in sampling rate. Based on the previous

learning and LSTM techniques. In the variable sampling rate case, the sampling rate that determines the accuracy of traffic prediction can change in real time with the dynamic changes of network traffic. Therefore, compared with the traditional traffic prediction methods based on fixed sampling rate, the traffic prediction method proposed in this paper can more accurately reflect the real-time changes of network traffic to further guide the reasonable and efective allocation of network resources.

analysis in Section [4.](#_bookmark7)1, the average sample rate *R*avg in VSR-

*VSR*

In the next work, based on the traffic prediction method

LSTM is lower than the fixed sample rate in FSS-LSTM. For with variable sampling rate proposed in this paper, we will

example, if we suppose the *ff* in the traffic load curve is further investigate the intelligent distributed allocation and

twice the *ff*

max and also assume *Ts*

*f*

max

� *T* , then, according ([1),](#_bookmark5)

management of multidimensional resources for the diferent

avg

*R*

*VSR*

in VSR-LSTM is 25% lower than the fixed sample rate

demands on resources such as computation, communica-

in FSS-LSTM.

# Conclusions

In this paper, we propose a network traffic prediction method with a variable sampling rate using machine

tion, and storage in 6G MEC IoTs networks for diferent vertical applications in the future.

Furthermore, we should also try to combine our pro- posed algorithm with Markov chains, hidden Markov models, and other classic prediction theories to improve the prediction accuracy, computational complexity, and other

performances, especially on small granular flow perfor- mance, such as a packet, a mobile app, etc. Meanwhile, we also try to extend the application of the novel prediction algorithm to new scenarios such as 6G, digital twin, and metaverse in the future.

# Data Availability

The dataset used to support the findings of the study can be obtained from a private company.

# Conflicts of Interest

The authors declare that they have no conflicts of interest.

# References

1. I. Khan, X. Tao, G. M. S. Rahman, W. U. Rehman, and

T. Salam, “Advanced energy-efficient computation offioading using deep reinforcement learning in MTC edge computing,” *IEEE Access*, vol. 8, pp. 82867–82875, 2020.

1. X. Liu, J. Yu, J. Wang, and Y. Gao, “Resource allocation with edge computing in IoT networks via machine learning,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3415–3426, April 2020.
2. M. Merluzzi, P. D. Lorenzo, and S. Barbarossa, “Wireless edge machine learning: resource allocation and trade-ofs,” *IEEE* *Access*, vol. 9, pp. 45377–45398, 2021.
3. J. Han, G. H. Lee, S. Park, and J. K. Choi, “Joint subcarrier and transmission power allocation in OFDMA-based WPTsystem for mobile edge computing in IoT environment,” *IEEE In-* *ternet of Things Journal*, vol. 9, no. 16, pp. 15039–15052, 2022.
4. Z. Chen, J. Hu, X. Chen, J. Hu, X. Zheng, and G. Min, “Computation offioading and task scheduling for DNN-based applications in cloud-edge computing,” *IEEE Access*, vol. 8, pp. 115537–115547, 2020.
5. P. W. Khan, K. Abbas, H. Shaiba, A. Muthanna,

A. Abuarqoub, and M. Khayyat, “Energy efficient computa- tion offioading mechanism in multi-server mobile, edge computing—an integer linear optimization approach,” *Elec-* *tronics*, vol. 9, no. 6, p. 1010, 2020.

1. I. Lohrasbinasab, A. Shahraki, A. Taherkordi, and A Delia Jurcut, “From statistical- to machine learning based network traffic prediction,” *Transactions on Emerging Telecommuni-* *cations Technologies*, vol. 33, no. 4, Article ID e4394, 2022.
2. T. Zhang, L. Feng, P. Yu, S. Guo, W. Li, and X. Qiu, “A handover statistics-based approach for cell outage detection in self-organized heterogeneous networks,” in *Proceedings of the 15th IFIP/IEEE International Symposium on Integrated Net- work and Service Management (IM ‘17)*, IEEE, Lisbon, Por- tugal, pp. 628–631, May 2017.
3. Y. Wang and T. Nakachi, “Prediction of network traffic through light-weight machine learning,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1919–1933, 2020.
4. Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, “Deep learning with long short-term memory for time series pre- diction,” *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114–119, 2019.
5. H. D. Trinh, L. Giupponi, and P. Dini, “Mobile traffic pre- diction from raw data using LSTM networks,” in *Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications* *(PIMRC)*, pp. 1827–1832.
6. G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescape, “Characterization and prediction of mobile- app traffic using Markov modeling,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 907–925, 2021.
7. A. Montieri, G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, “Packet-level prediction of mobile-app traffic using multitask deep learning,” *Computer Networks*, vol. 200, no. 2021, Article ID 108529, 2021.
8. S. Zeb, M. A. Rathore, A. Mahmood, S. A. Hassan, J. W. Kim, and M. Gidlund, “Edge intelligence in softwarized 6G: deep learning-enabled network traffic predictions,” in *Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps)*, De- cember 2021.
9. T. Ko, S. M. Raza, D. T. Binh, M. Kim, and H. Choo, “Network prediction with traffic gradient classification using convolu- tional neural networks,” in *Proceedings of the 2020 14th In- ternational Conference on Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–4, Tai- chung, Taiwan, January 2020.
10. J. Kwon, D. Jung, and H. Park, “Traffic data classification using machine learning algorithms in SDN networks,” in *Proceedings of the 2020 International Conference on Infor- mation and Communication Technology Convergence (ICTC)*, pp. 1031–1033, Jeju, Korea, October 2020.
11. H. Lu and F. Yang, “Research on network traffic prediction based on long short-term memory neural network,” in *Pro- ceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 1109–1113, Chengdu, China, December 2018.
12. J. Fan, D. Mu, and Y. Liu, “Research on network traffic prediction model based on neural network,” in *Proceedings of the 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pp. 554–557, Dalian, China, September 2019.
13. H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and

K. Wu, “Artificial-intelligence-Enabled intelligent 6G net- works,” *IEEE Network*, vol. 34, no. 6, pp. 272–280, November/ December 2020.

1. B. Rong, S. Sun, and M. Kadoch, “Traffic prediction for re- liable and resilient video communications over multi-location WMNs,” *Journal of Network and Systems Management*, vol. 24, no. 3, pp. 516–533, 2016.
2. S. Wu, W. Mao, C. Liu, and T Tang, “Dynamic traffic pre- diction with adaptive sampling for 5G HetNet IoT applica- tions,” *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–11, 2019.
3. Y. Zhao, “Highway traffic flow prediction based on simple recursive unit network,” *China’s transportation information*, no. 02, pp. 123–126, 2022.