Computer Networks 240 (2024) 110172

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An intelligent network traffic prediction method based on Butterworth filter and CNN–LSTM

Xueyan Hu, Wei Liu *, Hua Huo

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

ARTICLE INFO

Keywords:
Network traffic prediction
Butterworth filter
Long short-term memory network(LSTM)
Convolutional neural network(CNN)

ABSTRACT

Accurate and real-time network traffic prediction is of paramount importance in the fields of network management, performance optimization, and fault diagnosis. It provides strong support for autonomous network control, network administration and network services. Therefore, we propose a novel approach for network traffic prediction, which integrates the Butterworth filter, Convolutional Neural Network and Long Short-Term Memory network(BWCL). First, this method the network traffic data to frequency domain processing, utilizing the Butterworth filter to extract its low-frequency component. The residual component is generated by subtracting the low-frequency component from the network traffic sequence. Then, CNN–LSTM prediction models are employed to capture the spatial and temporal features of the data in different frequency bands. Finally, the prediction results of the two models are linearly summed to represent the final prediction value. To validate the feasibility of the proposed model, we construct a variety of datasets with statistical features by taking the raw network traffic in single\multi-cell scenarios at two different temporal granularities: minutes and hours. In the Pytorch experimental environment, we evaluate the performance of the model using MSE, $RMSE$, MAE, and R^2 performance metrics. The experimental results show that the prediction accuracy of the model is improved by 25% compared to the existing time series prediction models. This innovative approach provides new ideas in the field of time series forecasting, which has a broad application prospect.

1. Introduction

With the rapid development of information technology, the popularity of mobile Internet communication has reached an unprecedented level [1]. This trend has resulted in a huge scale of data generation. According to data released by the national bureau of statistics (NBS), mobile internet access traffic is up to 2618 trillion bytes in 2022, which is a significant year-on-year increase of 269% compared to 2018 http://www.stats.gov.cn/. The substantial growth in data traffic is closely associated with factors such as the increasing intelligence of Internet devices, the emergence of virtual reality technology, and the development of connected vehicles. The surge in data traffic compels enterprises and network providers to more efficiently manage and plan network resources to meet the urgent demands of users for high traffic and low latency. In this context, network traffic prediction is critical. Through the prediction of network traffic data, enterprises, and network service providers can anticipate potential traffic trends, thereby more effectively addressing the continuously growing demands. This not only makes resource allocation more forward-looking and improves bandwidth utilization efficiency but also helps develop strategic routing strategies. As a result, network traffic prediction has become an important tool to ensure high-quality service delivery and timely fulfillment of user demands [2].

In recent years, network traffic data prediction has been widely used in a variety of engineering scenarios, including intelligent substations [3], traffic speed [4], and software defined network(SDN) [5]. Through accurate traffic prediction and analysis, enterprises can better fulfill user expectations, provide efficient network quality of service, and reduce network congestion and resource wastage [6]. This ensures continuous improvement in network smoothness and superior user experience. At present, many classical statistical linear forecasting models have been widely used for their applications, such as mono-exponential smoothing [7], autoregressive (AR) model [8], autoregressive moving average (ARMA) model [9], and autoregressive integrated moving average (ARIMA) model [10]. However, with the widespread access of mobile terminals, network traffic data often exhibits unstable characteristics such as nonlinearity and randomness [11]. Therefore, the constructed prediction model must rely on more powerful data feature extraction capability. Deep learning has become the mainstream solution in the current direction of network traffic prediction with

* Corresponding author.
E-mail address: 990284@haust.edu.cn (W. Liu).

https://doi.org/10.1016/j.comnet.2024.110172
Received 17 August 2023; Received in revised form 21 December 2023; Accepted 4 January 2024
Available online 6 January 2024

its excellent data feature extraction capability. J.Guo, Z.Xie [12] introduced a prediction algorithm integrating support vector regression (SVR) and long short-term memory (LSTM) for the problem of predicting abnormal passenger flow data at stations of urban transportation networks. A.Thaduri et al. [13] proposed a deep learning framework based on convolutional neural network(CNN) for extracting local spatial features. S.Nihale [14] et al. proposed that the LSTM model not only compensates for the shortcomings of recurrent convolutional networks but also effectively captures the long-term dependence of data sequences.

Although the above models have made some progress in improving the overall prediction accuracy, they have not taken into account the prediction of residual component in traffic data. Network traffic data typically contains a great deal of noise, volatility, and uncertainty. Focusing only on the overall network traffic data and ignoring the predictive analysis of residual component will fail to accurately and timely capture the fluctuations and trends in the traffic data. Therefore, this paper proposes an innovative network traffic prediction method that combines the Butterworth filter [15], CNN, and LSTM neural network models. We use a Butterworth filter to smooth the traffic data and obtain the low-frequency component and the residual component. CNN is used to capture the spatial features of traffic data in different frequency bands. LSTM compensate the problems of gradient explosion and gradient vanishing in traditional recurrent neural network (RNN), and captures the long-term dependencies of time series data more effectively. This approach successfully overcomes the limitations of traditional methods in dealing with nonlinear and non-stationary data. In addition, we construct CNN–LSTM prediction models to adapt different frequencies to effectively capture the spatio-temporal characteristics of low-frequency and residual data. Overall, we build a comprehensive and accurate network traffic prediction model, and experimentally confirm and evaluate the method. Therefore, the main contributions of this paper are summarized as follows.

- We propose a network traffic data prediction model based on BWCL. This pioneering approach utilizes a Butterworth filter to preprocess the raw data and obtain the low-frequency component and the residual component. Then, separate predictions are made for each component. Finally, the predictions are combined to improve the accuracy and reliability of the overall prediction.
- We construct different CNN–LSTM hybrid prediction models to target different frequency bands of Butterworth output with multiple features. By combining different data characteristics with the appropriate prediction model, the prediction performance is greatly improved compared with the traditional single prediction model.
- To meet the needs of different application scenarios, we conduct prediction at two different time granularities: minute-level and hour-level. The minute time granularity is suitable for real-time and high-frequency monitoring, while the hour granularity provides a more macroscopic summary and trend analysis.
- We train and predict the mean, minimum, and peak values of raw data by different BWCL hybrid models to gain a deeper understanding of the overall trend and fluctuation of network traffic data. It helps to rationally plan resources and optimize performance to cope with traffic bursts. This approach provides comprehensive and accurate insights to better understand and predict the dynamics of time series data.

We organize the rest of the paper as follows. Section 2 introduces the theory of the relevant network models involved in this thesis and their concepts. Hybrid models of deep neural networks are described in detail in Section 3. In Section 4, we show the experimental results as well as the performance evaluation of the model and further provide a full discussion and analysis of the experimental results. Finally, we summarize our work and outline prospects in Section 5.

2. Related work

The definition of network traffic is the volume of data transmitted within a computer network, representing the overall quantity of data that flows through a network device or transmission medium during a specified sampling period [16]. In real-world scenarios, due to the complexity of internet applications, the diverse dynamics of user lines and the occurrence of sudden network events, a significant amount of network traffic emerges within extremely short time, which exhibits a variety of nonlinear characteristics such as abrupt changes, trends and periodicity [17]. These unique characteristics give network traffic accurate prediction extremely challenging and have attracted extensive academic attention. Currently, network traffic prediction methods can be broadly categorized into two main groups: model-driven and data-driven approaches.

The model-driven approach, also known as the parameter model method, achieves an approximate fit to network traffic sequences through the adjustment of its numerous parameters. Q T.Tran et al. [18] applied exponential smoothing as a method for adaptive prediction of speech and data, and evaluated the prediction accuracy of different exponential smoothing models. C.Bhar et al. [19] predicted the filling time of optical network unit (ONU) buffers using an ARMA model. This enabled the adoption of various power-saving modes to minimize energy consumption. J.Lee et al. [20] employed photovoltaic generation data and gold price data to train online ARIMA models with different optimizers. Experimental results demonstrate that identifying the optimal optimizer based on the characteristics of the dataset is a crucial element for enhancing prediction accuracy. To further improve its overall prediction accuracy. L.Deng, K.Ruan et al. [21] used ARIMA and neural basis expansion analysis for interpretable time series forecast (N-BEATS) to construct the hybrid model for predicting IP network traffic, N-BEATS predicted the residual values of the ARIMA model, the sum of the ARIMA prediction and its prediction residuals was used as the final prediction result. However, all of these methods mentioned above use a polynomial fit function network to approximate their network traffic sequences, which makes it difficult to capture the nonlinear characteristics of the time series data.

Data-driven refers to the approach of guiding decisions and actions through data collection, analysis, and application, enabling intelligently analyze and capture the correlation and non-linear properties of data. Specifically, data-driven can be categorized into machine learning and deep learning. S. Zhang, M. Han et al. [22] proposed a sliding window based on a sparse kernel recursive least squares algorithm (SW-SKRLS) that effectively improves the prediction in time-varying environments. A, Xiushan Jiang et al. [23] effectively and accurately predicted the short-term passenger flow of high-speed railroad (HSR) based on an integrated empirical mode decomposition (EEMD) and gray support vector machine (GSVM) hybrid model. L. Nie [24] used the reinforcement learning (RL) model to predict end-to-end network traffic prediction in the smart Internet. Additionally, a Monte-Carlo Q-learning algorithm was proposed to address the issue of high resource consumption resulting from large-scale network traffic. Deep learning, as a data-driven approach, can efficiently handle massive amounts of data with powerful automatic feature extraction and pre-processing capabilities [25]. Therefore, they have a natural advantage in time series modeling tasks and are used in different industrial scenarios. For example, solar photovoltaic power prediction [26], traffic prediction [27], and base station load prediction [28]. N.Ranjan et al. [29] employed a neural network prediction model that integrates CNN, LSTM, and Transpose convolutional ceural cetwork (Transpose CNN) to effectively predict end-to-end traffic congestion in transportation networks. J. Fan, D. Mu, et al. [30] applied a composite model combining deep RNN and gated recurrent unit (GRU) neural networks to network traffic prediction. Z. Zhao [31] studied the CNN–LSTM road condition prediction model based on spatio-temporal trajectory topology map to solve the road condition prediction problem for cab

Table 1
Full name of the different models.

Model abbreviations	Full model name
BW	Butterworth Filter
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory network
BWCL	Butterworth filter and Convolutional Neural Network and Long Short-Term Memory Network.
RNN	Recurrent Neural Network
SDN	Software-Defined Network
AR	Autoregressive
ARMA	Autoregressive Moving Average
N-BEATS	Neural Basis Expansion Analysis for Interpretable Time Series Forecast
SW-SKRLS	Sparse kernel Recursive Least Squares Algorithm based on Sliding Windows
GCN-GRU	Graph Convolutional Networks-Gated Recurrent Unit
EEMD	Empirical Mode Decomposition
GSVM	Gray Support Vector Machine
HMM	Hidden Markov Model
RL	Reinforcement Learning

data. S. Huaifeng et al. [32] proposed a joint attention and GCN-GRU method for network traffic prediction, whose model achieved better prediction results with different performance metrics. Y. Pen et al. [33] improved the wireless network traffic prediction accuracy by obtaining its hidden state information as training input data for LSTM through hidden Markov model (HMM). Y. He et al. [34] proposed a multi-channel spatial–temporal graph convolutional network to solve the problems of single data dimension and spatially correlated features of different nodes, which largely improved its prediction accuracy. H.Zhou,et al. [35] proposed the prediction model named Informer for the long sequence time prediction problem and utilized probabilistic sparsity to effectively deal with the memory usage as well as time complexity issues. The full names of all the above models are shown in Table 1

Unlike the previous studies, we innovatively combine the Butterworth filter, CNN, and LSTM. From the perspective of signal processing, Butterworth filter is utilized to divide the data into different frequency bands. Following this, the prediction analysis is performed by the CNN–LSTM neural network model. In this two-stage process, the respective advantages of both are fully utilized, further improving the network traffic prediction accuracy. This innovative approach has important application potential in the field of network traffic prediction, providing strong support for improving network performance and security.

3. The proposed approach

We propose a BWCL model, which not only utilizes the excellent smoothing and denoising of data by Butterworth filter but also integrates CNN and LSTM to extract the spatial–temporal features of time series. Compared with traditional prediction models, it has better data processing and prediction capabilities in cell network traffic prediction. In this section, the dataset used is described and partitioned, followed by an overview of the overall forecasting framework for the time series. Finally, the Butterworth filter and the space–time modeling algorithm based on CNN–LSTM are given in detail.

3.1. Dataset description and analysis

3.1.1. Dataset introduction

The cellular traffic dataset analyzed in this paper is from an open dataset of Italian network traffic http://dataverse.harvard.edu/dataset.xhtml?persistentId=doi.org:10.7910/DVN/EGZHFV. The data collection area of the entire Milan city is divided into 100*100 grid covers, each square is about 235 × 235 meters in size, which we call cells.

Table 2
Dataset table.

Dataset	Milan telephone service provider
City	Milan
Country code	39
Time span	2013/11/01-2014/01/01
Time interval	10 min
Grid size	(100,100)

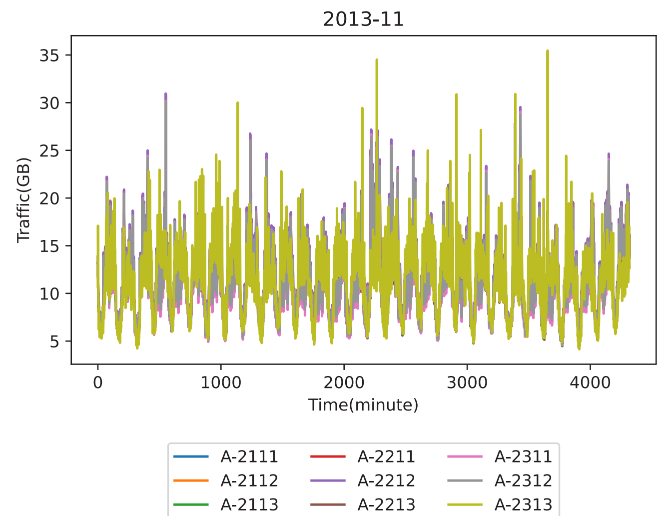

Fig. 1. Trends in network traffic data for nine regions in November 2013.

This dataset collects three types of cellular traffic in the time interval from 00:00 on November 1, 2013 to 00:00 on January 1, 2014, namely SMS transmission, inbound and outbound calls, and network traffic activities. In this experiment, only network traffic data with city code 39 is selected as the experimental dataset with a network traffic data time interval of 10 min as well as 144 data points are collected per day for each region. We divide it into several sub-datasets according to different prediction duration requirements, the related dataset information is given in Table 2. Fig. 1 shows in detail the trend of network traffic data in November 2013 for nine areas (A-2111∼A-2113, A-2211∼A-2213, A-2311∼A-2313) of the selected city code 39.

3.1.2. Build datasets

To address the different reliance on long/short-term datasets for various network traffic prediction scenarios. We construct two datasets with different temporal granularity, namely, minute and hour. Specifically, the minute-time granularity dataset provides higher temporal resolution, enabling more detailed observation and capture of short-term fluctuations and changes in traffic. This is important for application scenarios that require real-time and high-frequency monitoring, such as financial trading systems or real-time network monitoring. Hourly time granularity, on the other hand, provides a smoother summary of the data, which can reduce the impact of noise and outliers, provide reliable support in global data analysis and long-term change trends. It lends itself well to macro level traffic analysis. Different time granularities are selected to predict for different application scenarios, reducing the computational burden of the prediction model while maintaining excellent prediction performance.

Considering the differences in prediction model complexity together and dataset scale required for the two different time granularities, the dataset is divided as well as processed accordingly. Firstly, in the minute time granularity, we select the data with the interval 2013/11/01∼2013/11/07, the time step of 10 min as training and validation datasets of the hybrid prediction model. Second, considering

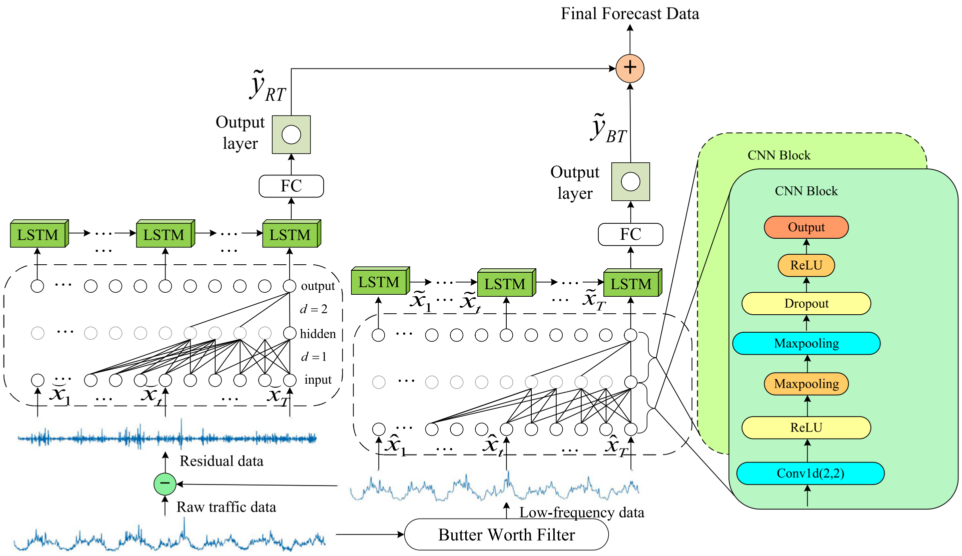

Fig. 2. The overall model framework diagram.

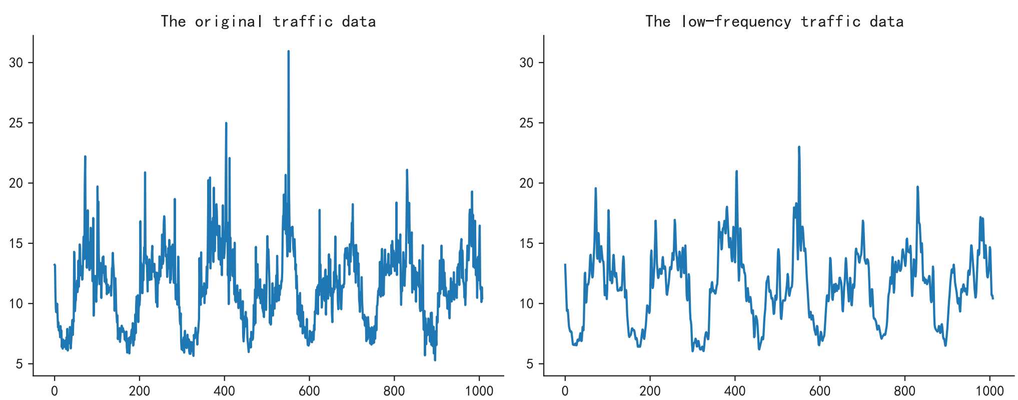

Fig. 3. 2013/11/01~2013/11/07 data visualization.

the sparsity of the hourly time granularity, we use all the data for the subsequent simulation experiments. Compared to the minute time granularity prediction, the hourly time granularity requires consideration of how to select the appropriate data values for this hour. Therefore, the training and validation sets are reconstructed into three sets with different statistical characteristics of the data, namely mean, minimum and peak values in each hour. Assuming that the data used for one hour are $\mathbf{x} = \{x_n\}_{n=1}^{6}$.

- The mean is a measure of the overall trend of traffic data. By calculating the mean value of the flow data, understanding of the average level of the data can be obtained and used to assess the accuracy together with degree of deviation of the model prediction results.

$$\bar{x} = \frac{1}{6}\sum_{n=1}^{6} x_n \tag{1}$$

- Peak is the highest traffic value within a certain time range, reflecting the burst and high load of network traffic. In the prediction model, accurate prediction of peak traffic is important for reasonable planning of network capacity and bandwidth, as well as network resource allocation.

$$x_{\max} = \max\{x_n\}_{n=1}^{6} \tag{2}$$

- The minimum value is the lowest traffic value within a certain time range, reflecting the idle and low-load state of the network. Accurately predicting the minimum traffic can help network administrators in resource planning and performance optimization, so as to avoid wasting resources while meeting the demand during low load.

$$x_{\min} = \min\{x_n\}_{n=1}^{6} \tag{3}$$

Considering the different correlation features between single cell and between multiple cells, we choose A-2212 as the target prediction cell to train and predict two different time-granularity datasets in single/multi-cell scenarios, respectively. Precisely, the single-cell prediction uses only the network traffic data of cell A-2212 to build hybrid BWCL prediction model, which is more focused on the specific features and periodic changes within the cell and easier to model and predict. For multi-cell scenarios, the prediction model needs to take into account the correlation between cells, data interaction, spatial correlation and cell heterogeneity. Although it requires more complex model structure and parameter tuning, the model has stronger generalization capability and can more accurately capture the overall network traffic trends, providing more valuable information for network management, performance optimization and resource provisioning decisions.

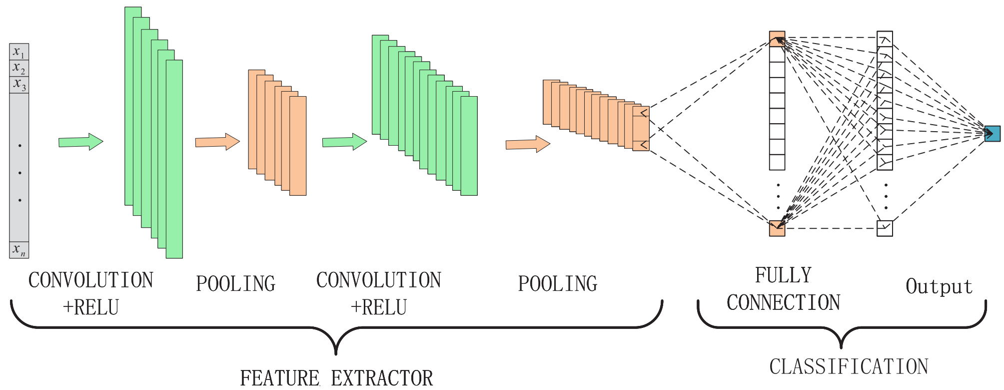

Fig. 4. 1D CNN schematic.

3.2. Overall framework of $BWCL$ model

We denote the length of original time series T with time span $t \in \{10\,\mathrm{min}, 1\,\mathrm{h}\}$ as $\mathbf{x} = \{x_1, x_2 \cdots x_T\}$ which is fed into the model for training and prediction, as shown in Fig. 2. Firstly, We utilize Butterworth filters for effective denoising and smoothing of the original time series data, filtering out its high-frequency components. We obtain a low-frequency signal $\hat{\mathbf{x}} = \{\hat{x}_1, \hat{x}_2 \cdots \hat{x}_T\}$ with the same time span T. Next, we introduce the CNN module for capturing local spatial features $\tilde{\mathbf{x}} = \{\tilde{x}_1, \tilde{x}_2 \cdots \tilde{x}_T\}$ in the low-frequency data $\hat{\mathbf{x}} = \{\hat{x}_1, \hat{x}_2 \cdots \hat{x}_T\}$ enables the model to better understand the spatial distribution characteristics of time series data. Then, local spatial features $\tilde{\mathbf{x}} = \{\tilde{x}_1, \tilde{x}_2 \cdots \tilde{x}_T\}$ serves as the input of the LSTM module, processing its time dimension information. The memory unit in LSTM can preserve historical time information, which can effectively capture temporal long-term dependencies in data $\tilde{\mathbf{x}}$. Finally, the prediction result $\tilde{y}_{BT}$ ($\tilde{y}_{BT}$ denoting the low-frequency signal prediction value) is obtained through the output of the fully connected layer. Spatially local features and global temporal features are comprehensively considered in this mutual collaboration approach, which leads to a more comprehensive analysis of the time series data, and improves the prediction accuracy and robustness for the time series data.

In addition, to ensure the completeness of the flow data, and further improve the overall prediction accuracy, we must consider the analysis and prediction of the residual data. As shown in Fig. 2, the residual signal $\overline{\mathbf{x}} = \{\overline{x}_1, \overline{x}_2 \ldots \overline{x}_T\}$ is obtained by subtracting the low-frequency data $\hat{\mathbf{x}}$ from the time-series data $\mathbf{x}$, which has random oscillation characteristics and unstable qualities such as nonlinearity. Therefore, we construct a hybrid neural network prediction model different from the low-frequency data for separate training and prediction. The same process as described above for the prediction of low-frequency signals is used to obtain $\tilde{y}_{RT}$ ($\tilde{y}_{RT}$ denotes the residual signal prediction value). Finally, we linearly sum the predicted values of the low-frequency and residual signals, thereby generating the ultimate forecasting result $\tilde{y}_T$.

In experiments, we use historical data from the past T time span to predict the data values at the moment $T+1$, and then find the best nonlinear mapping function from input data to predict values by minimum prediction error of the neural network model. We define the nonlinear mapping function as $\tilde{y}_T$ and name the overall prediction model as $BWCL(\mathbf{x})$, where $\mathbf{x}$ is the original time series data.

$$\tilde{y}_T = BWCL(\mathbf{x}) \tag{4}$$

3.3. Butterworth filter

The Butterworth filter is a classical filter with excellent frequency response characteristics, commonly used for smoothing time domain signals and removing high-frequency noise. There are often certain network fluctuations and interference within cells and between cells, which makes the collected network traffic time series data adulterated with high-frequency noise and bursty outliers, reducing the accuracy of predicting the trend of future network traffic data changes. By taking advantage of the property that the frequency response curve of Butterworth is maximally flat in the passband. The overall trend of the data can be retained while effectively removing high-frequency noise and sudden outliers from the flow data. To more visually observe the ability of the Butterworth filter to retain the overall variation trend and direction. We visualize 2013/11/01∼2013/11/07 as shown in Fig. 3. Therefore, processing the network traffic data through the Butterworth filter first is the key to improving the prediction ability.

The specific data processing process is as follows, we first determine the optimal Butterworth filter cutoff frequency w_c and order N by combining network traffic data characteristics of multiple cells, complete the construction of the Butterworth filter, whose transfer function is expressed as.

$$H(jw) = \frac{1}{1 + (\frac{jw}{w_c})^{2N}} \tag{5}$$

We take the original network traffic as its input signal $x(n)$ through discrete time Fourier transform (DTFT) to get $X(jw)$, and multiply it with the transfer function $H(jw)$ to get the frequency domain representation of the filter output data in the form of $\hat{X}(jw)$, finally get the time domain representation of output signal in the form of $\hat{x}(n)$ through discrete time Fourier inverse transform (IDTFT), and the data processing equation is shown below.

$$\hat{X}(jw) = X(jw)H(jw) \tag{6}$$

$$\hat{x}(n) = IDTFT[\hat{X}(jw)] \tag{7}$$

Where, $\hat{x}(n)$ represents the filtered low-frequency data, which contains the two main components of long-term linear growth and periodic changes of network traffic, so that network traffic has certain characteristics of continuity, regularity and smoothness, the low-frequency data $\hat{x}(n)$ often also has certain mathematical statistical laws. These characteristics make it easier to construct hybrid prediction models and ensure the feasibility and accuracy of low-frequency data prediction. We use the original network traffic $x(n)$ minus $\hat{x}(n)$ to obtain the residual data $\overline{x}(n)$ of the network traffic by a simple linear operation.

$$\overline{x}(n) = x(n) - \hat{x}(n) \tag{8}$$

Compared to low-frequency data $\hat{x}(n)$, $\overline{x}(n)$ reflects the random oscillations and irregularities of network traffic, these characteristics make predicting $\overline{x}(n)$ more challenging. To better fit $\overline{x}(n)$, we need to construct more complex mixture models to capture its nonlinear and irregular features. For low-frequency and high-frequency data, we need to construct hybrid prediction models with different levels of complexity to ensure the prediction accuracy of each component while ensuring prediction accuracy of the overall data.

3.4. CNN-LSTM model

For the spatial feature extraction problem, which is currently one of the keys to network traffic prediction, we introduce the popular neural network CNN, also known as feedforward neural network, in our prediction model, which has excellent spatial feature capture capability. As shown in Fig. 4: it is mainly composed of convolutional layer, pooling layer and fully connected layer, while the introduction of dropout layer is usually considered in the face of overfitting problem. We use 1DCNN pair of low-frequency time series data that pass through Butterworth low-pass filter, whose input is $l \in R^T$ and the convolution layer is $f : \{0, 1, \ldots, k-1\} \rightarrow R$, whose convolution function is defined as.

$$G(l_t) = (l * f)(t) = \sum_{i=0}^{k-1} f_i l_{t-i} \tag{9}$$

$$seqout = (G(l_1), G(l_2), \ldots, G(l_T)) \tag{10}$$

where k is the convolution kernel size, $G(\cdot)$ is the convolution function, $seqout$ is the time series output by the convolution operation.

It is well known that although CNN has a strong ability to capture the spatial features of data, it is significantly deficient in extracting the long-term dependence of time series information. While LSTM, one of the variants of RNN, has a natural advantage in capturing the long-term dependence of cellular traffic series information. We use the local spatial features $\tilde{\mathbf{x}} = \{\tilde{x}_1, \tilde{x}_2 \cdots \tilde{x}_T\}$ as inputs to the LSTM, and learn the nonlinear mapping function between $\tilde{\mathbf{x}}$ and H_t at the moment t

$$H_t = \sigma(H_{t-1}, \tilde{X}_t) \tag{11}$$

H_t and H_{t-1} are the hidden states of the LSTM at moments t and $t-1$, respectively. $\sigma(\cdot)$ denote the activation functions.

LSTM mainly consists of input gate (i), forget gate (f) and output gate (o), which control the flow of information. The forgetting gate determines the proportion of forgotten information in the cellular state at the previous moment in time. The input gate controls the extent to which input information is added to the memory cell state at the current moment. Finally, the output gate determines the amount of information output from the cell state at the current moment. Among them, the memory cell (C) the most central component of the LSTM, is similar to a memory for storing and managing the information in the network. The memory cell allows the LSTM to remember information over time and pass it on to other time steps when needed. The formula for the LSTM unit from input to output is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{12}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{13}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{14}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{15}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{16}$$

$$h_t = o_t \odot \tanh(C_t) \tag{17}$$

where f, i and o represent forgetting gate, input gate and output gate at moment t, respectively; The values of the memory cell and candidate memory cell at the current moment are represented by C_t and $\tilde{C}_t$; For W and b with different subscripts, they are used to represent the corresponding weight matrix and bias vector; $\sigma(\cdot)$ and $\tanh(\cdot)$ denote

Table 3
Training parameters.

Training parameters	
Optimizer	Adam
Learning rate	0.001
Epoch	1000
Loss function	MSELoss
N	3
f_c	10 KHz
w_c	200 Hz

the sigmoid function and hyperbolic tangent function, respectively; $\odot$ denotes the point operation (element-by-element multiplication); h_t denotes the hidden state at the current moment; $[,]$ denotes the splicing of vectors.

3.5. Optimization of the BWCL model

The optimization objective of the BWCL network model proposed in this paper is mainly to find a set of parameters to minimize the prediction error. Therefore, we assume that the sample size is N and the cross-entropy loss function of the model is:

$$J(W) = -\frac{1}{N} \sum_{t-1}^{N} [y \ln(\tilde{y}) + (1-y) \ln(1-\tilde{y})] \tag{18}$$

Where, y is the actual flow data value; $\tilde{y}$ is the prediction result. The weight parameter matrix W is updated in combination with the Adam optimization algorithm to accelerate convergence to improve model performance:

$$W_{t+1} = W_t - \alpha \frac{m_t}{\sqrt{v_t} + \varepsilon} \tag{19}$$

In the equation, α represents the learning rate of the neural network, while m_t and v_t denote the first-order moment estimates (mean) and second-order moment estimates (variance) of the parameter gradient at the current moment, which can be expressed as:

$$m_t = \beta_1 \cdot m_{t-1} + \left(1 - \beta_1\right) g_t \tag{20}$$

$$v_t = \beta_2 \cdot v_{t-1} + \left(1 - \beta_2\right) g_t^2 \tag{21}$$

$$g_t = \nabla J(W) = \frac{\partial (J(W))}{\partial (W)} \tag{22}$$

β_1 and β_2 are Adam's first-order moment estimation and second-order moment estimation tuning parameters, respectively.$\nabla J(W)$ is the gradient of the loss function at point $J(W)$.

By calculating the cross-entropy loss function of the real network traffic data and the prediction results, the reverse transfer algorithm is used to fine-tune its target task model. And calculate whether the updated loss function meets the demand of current traffic data prediction accurately, the calculation formula is as follows:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\tilde{y}_\theta(x_i), y_i) + \lambda \sum_{j=1}^{n} \theta_j^2 \tag{23}$$

$J(\theta)$ is the fine-tuned loss function; $\tilde{y}_\theta(x_i)$ denotes predicted results; $\mathcal{L}(\tilde{y}_\theta(x_i), y_i)$ is the cross-entropy loss function of predicted values and the true data. is the regularization coefficient. Predictive model overfitting is reduced by calculating the sum of squares of the weights, which improves the prediction accuracy in the test dataset. The specific prediction and optimization process of the BWCL model is shown in the algorithm 1.

Algorithm 1 Overall Algorithm: BWCL based network traffic data prediction implementation process

Input: Target domain dataset $x(n)$
Output: Final forecast value $\tilde{y}_T$.
1: Butterworth lowpass$[x(n)] \rightarrow \hat{x}(n)$
2: $x(n) - \hat{x}(n) \rightarrow \breve{\mathbf{x}}(n)$
3: Split $x(n)$ by 8:2 to get $train\_x$,$train\_y$,$test\_x$,$test\_y$
4: Load the constructed CNN–LSTM model,input $\hat{x}(n)$ into the model
5: **for** epoch $i \in [1, 1000]$ **do**
6: $train\_x, train\_y \rightarrow train\_\tilde{y}$
7: Remove the gradient accumulation at each propagation
8: Calculate the loss $J(W)$ using cross-entropy loss function
9: $J(W) = -\frac{1}{N}\sum_{i=1}^{N}[y\ln(\tilde{y}) + (1-y)\ln(1-\tilde{y})]$
10: Reverse passes update the CNN–LSTM model parameters and compute the fine-tuned loss values $J(\theta)$
11: $J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\tilde{y}_\theta(x_i), y_i) + \lambda\sum_{j=1}^{n}\theta_j^2$
12: Judge whether $J(\theta)$ is the termination state, if yes, get the target task CNN–LSTM model, jump out of the loop
13: **end for**
14: CNN–LSTM $(test\_x) \rightarrow test\_y$
15: Load different CNN–LSTM model,input $\breve{\mathbf{x}}(n)$ preprocessed into the model.
16: Residual data as in the above steps(5-12)
17: CNN–LSTM $(test\_xr) \rightarrow test\_yr$
18: Calculate Final forecast value $\tilde{y}_T = test\_y + test\_yr$

4. Experimental design and comprehensive analysis of BWCL model

We simulate and analyze the constructed BWCL hybrid neural network model. First, we give the prediction performance of the constructed model under two different time granularities (single-cell minutes and hours). Subsequently, more complex multi-cell network traffic prediction scenarios are also fully simulated. Among them, MAE, MSE, $RMSE$ and R^2 are performance evaluation metrics for the proposed model. Finally, we expand on three aspects of model performance evaluation, environment construction, and experimental results.

4.1. Performance evaluation metrics

In this section, we set four performance evaluation metrics to evaluate the prediction performance of the BWCL model proposed in this paper. MAE, MSE, $RMSE$ reflects the exact error between the predicted future data values and the real values of the model. R^2 is mainly used to reflect the degree of fitting between the predicted data of the model and the actual observations. We measure the prediction accuracy and fitting effect of the proposed model in this paper from multiple perspectives.

- Mean Square Error (MSE) is a metric with a value range of $[0, +\infty)$ to evaluate the performance of neural network model, which can be used to measure the degree of difference between predict values and the sample values. Specifically, the performance of the model will improve as the MSE decreases. It can be expressed as:

$$MSE = \frac{1}{T}\sum_{i=1}^{T}[y(n)_i - \tilde{y}(n)_i]^2 \tag{24}$$

- The Root Mean Square Error ($RMSE$) is the square root of the MSE and is used to measure the standard deviation of the prediction error, which provides a more intuitive picture of the prediction error performance in terms of order of magnitude. The value range is $[0, +\infty)$, and the model performances will be enhanced as its value approaches zero. This formula is:

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}[y(n)_i - \tilde{y}(n)_i]^2} \tag{25}$$

- Mean Absolute Error (MAE) is used to measure the average of the distance between the predicted and true values. Compared to MSE, it is more friendly to outliers. Its value range is $[0, +\infty)$, and as the value is taken closer to zero, the more superior the performance of the model will be. The formula is:

$$MAE = \frac{1}{T}\sum_{i=1}^{T}\left|y(n)_i - \tilde{y}(n)_i\right| \tag{26}$$

- R^2 (Determination Coefficient) is used to measure the degree of fit of the prediction model. Its value range is (0,1), and the closer the degree of fit is to 1, the more accurate the predicted value is. The formula is:

$$R^2 = 1 - \frac{\sum_{i=1}^{T}[y(n)_i - \tilde{y}(n)_i]^2}{\sum_{i=1}^{T}[\bar{y}(n)_i - \tilde{y}(n)_i]^2} \tag{27}$$

where N denotes the number of samples, $y(n)_i$, $\tilde{y}(n)_i$ and $\bar{y}(n)_i$ denote the actual, predicted and average values of the flow data at the ith moment, respectively.

4.2. Experiment environment and settings

In the experiment, Python 3.9 is used as the programming environment for training the BWCL model with Pytorch to complete the construction of the overall network framework. The specific parameter settings for the BWCL model in question are shown in Table 3. For simulations at two different time granularities, minutes and hours, we set the time window L to 6 and 24, respectively. The former denoting the use of the current hour's data to predict the next ten minutes of data. The latter is used to denote the use of a full day's worth of current data to predict an hour's worth of data in the future. To reduce the complexity of neural network training and model computation burden while preserving the inherent patterns of the original data samples. We extract three different hourly granularity traffic datasets: maximum, minimum and mean values within each hour of the original dataset. The detailed analysis is described in Section 3.1

In addition, in the face of the adverse effects of traffic fluctuations in different cells and the problem of predicting samples at different time scales. We flexibly use a variety of hidden layers and hidden neurons and complete the construction of the neural network to adapt to different data samples.

4.3. Full experimental results and analysis

This experiment accomplishes the traffic prediction task for single/multi-cell scenarios at two temporal granularities, minutes and hours, respectively. We use 80% of the raw network traffic data as training data for its neural network model and the remaining 20% as the test set for the $BWCL$ model.

4.3.1. Analysis of single-cell minute prediction result

As shown in Fig. 5, the curves of the predicted and actual values of the low-frequency data for single cell (A-2212) almost overlap, which greatly validates the feasibility of the BWCL prediction model. The predictive performance parameters of the different models are given in Table 4. Among them, the MAE, MSE, $RMSE$, and R^2 of the low-frequency signals are 0.20101306, 0.07117987, 0.26679555, and 0.98988217, respectively, which further illustrates that the BWCL model has a good prediction performance for low-frequency data. Fig. 6 shows the prediction results for the residual signal. The prediction

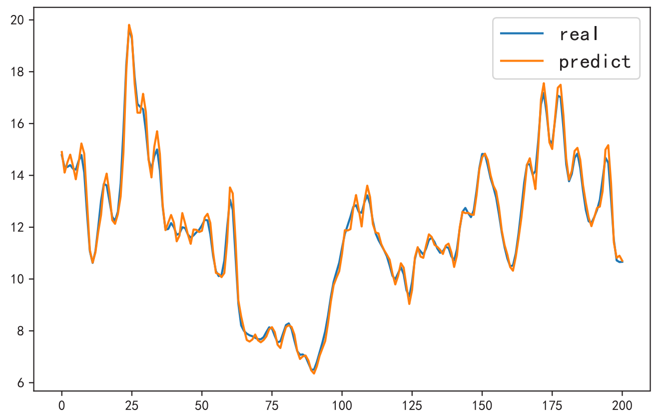

Fig. 5. Low-frequency signal prediction result.

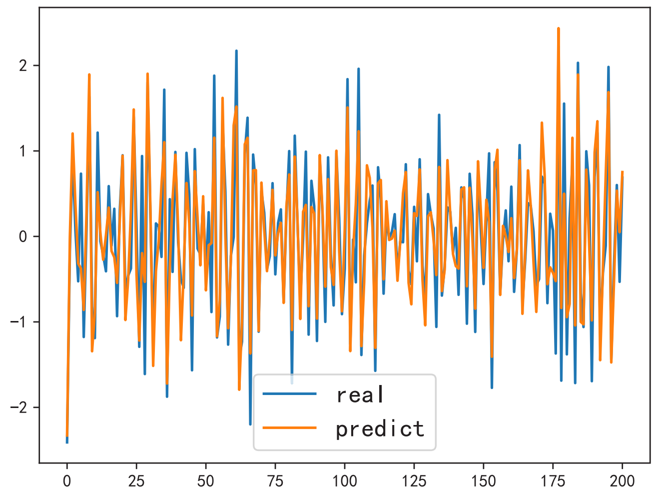

Fig. 6. Residual signal prediction result.

Table 4
Single cell minute prediction error performance table.

Data types	MAE	MSE	RMSE	R^2
Low-frequency data	0.20101306	0.07117987	0.26679555	0.98988217
Residual data	0.29557708	0.1471771	0.38363668	0.82338244
CNN	1.2067419	2.481206	1.5751845	0.6890737
CNN–LSTM	1.2135938	2.4557416	1.5670806	0.6922648
BWCL	0.3548058	0.2264362	0.4758531	0.9716247

accuracy is slightly off relative to the low-frequency data where trend smoothing has a periodic character. This is because residual data usually exhibits uneven distribution, noise, and random oscillations, which results in lower data quality. These properties make it difficult for the prediction model to adequately capture the key feature information in the residual data, resulting in prediction accuracy being slightly lower than the low-frequency signal. To solve the above problems, we consider key factors such as model complexity, learning rate, and number of neurons, constructing a prediction model distinct from the low-frequency signal. In Table 4, the MAE, MSE, $RMSE$, and R^2 of the residual signal are 0.29557708, 0.1471771, 0.38363668 and 0.82338244, respectively. It can be seen that the BWCL model also has a good advantage in residual signal prediction.

To verify the change in prediction performance brought about by the Butterworth filter in the proposed model after processing raw data in the frequency domain. As shown in Fig. 7, we conduct predictive analyses on raw network traffic data using the baseline CNN and the CNN–LSTM models. Because the experiments focus on univariate, short-term time series forecasts without more complex spatio-temporal characteristics, the computation time of the three forecasting models is relatively close (about 17 s). However, the prediction accuracy of the BWCL model is 40% higher than that of the baseline CNN and CNN–LSTM models. Fig. 8 can more intuitively see the prediction advantages of BWCL. This prediction result also verifies that it is feasible for the BWCL model to predict low-frequency and residual signals individually, and sum them up as the final prediction result. It not only improves the prediction accuracy of network traffic data but also ensures the integrity of the data.

4.3.2. Analysis of single-cell hourly prediction results

The experimental results of the different model for the minimum, mean, and peak values of the network traffic at the hourly time granularity of a single cell are shown in Figs. 9–11. Table 5 provides the corresponding prediction error performance parameters. Fig. 12 can more intuitively show the error performance gap between CNN, CNN–LSTM, and BWCL models.

As shown in Figs. 9–11, due to the influence of factors such as the complex spatio-temporal characteristics of single-cell hourly time granularity, data quality, and noise. CNN and CNN–LSTM are unable to adequately capture time series features, leading to their low prediction accuracy. The BWCL prediction performance is to some extent far better than the CNN–LSTM, which is also more intuitively verified in Fig. 12. In Table 5, the mean $RMSE$ and R^2 of the BWCL model are 0.5671231 and 0.9824678, respectively. The $RMSE$ and R^2 of the minimum values are 0.5636718 and 0.9807507, respectively. Compared with the CNN and CNN–LSTM prediction models, the computational efficiency of BWCL is slightly reduced by about 10 s, but the prediction accuracy is improved by more than 18.07% and 12.64%. Compared to CNN–LSTM peak prediction, due to the frequency and random oscillatory nature of the peak data itself. The $RMSE$ of BWCL decreases from 2.4658544 to 0.7670608, and R^2 increases from 0.7565565 to 0.9764428, which improves its prediction accuracy by 29.10%. MAE and MSE error performance indicators are also significantly reduced. Therefore, although the computational efficiency of this model is slightly lower than that of CNN and CNN–LSTM prediction models, it achieves accurate prediction when dealing with different data and improves the prediction accuracy of each attribute data.

Fig. 13 shows the prediction curves of the BWCL neural network model for the mean, peak, and minimum data compared to the real data. We can see the amplitude range after the prediction of different attribute data. It not only reflects the general trend of future changes in network traffic data but also assists in determining the load condition of network traffic. Specifically, when the prediction results are close to the peak, the network is under high load. We can avoid network congestion by adjusting the network capacity and bandwidth. On the other hand, when the prediction result is close to the minimum value, the network is in a relatively idle or low-load state, to avoid the waste of network resources, we can optimize the allocation of network resources or adjust the working mode of the base station in time.

4.3.3. Analysis of multi-cell minute prediction results

The above experiments are all based on training predictions for a single cell (A-2212). Since the single variable does not fully consider the impact of network traffic in neighboring cells on the target cell in terms of space and time, its prediction results have low confidence. Therefore, the following experiments are mainly focused on multi-variable single-output predictions for multiple cells. We collect network traffic data from nine cells ($2311 \sim 2313, 2211 \sim 2213, 2111$ ~2113), and use (A-2212) as predictive label. Fig. 14 shows the prediction results of CNN, CNN–LSTM, and BWCL models. The corresponding prediction error performance is given in Table 6.

Combining the error performance data in Fig. 14 and Table 6, we can observe that the CNN and CNN–LSTM prediction models are less

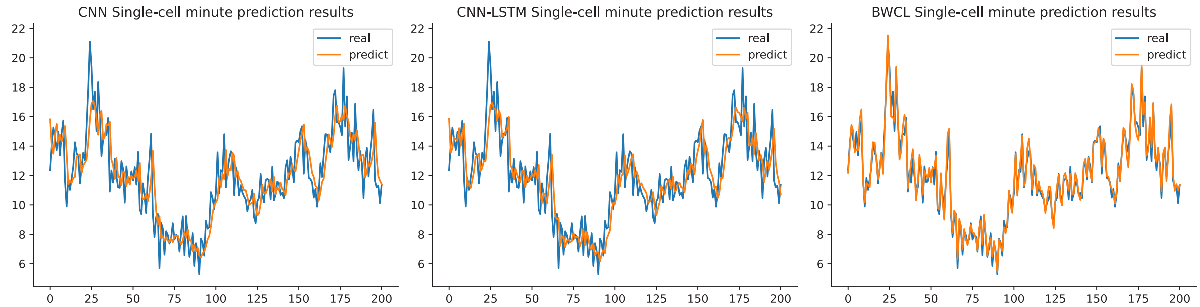

Fig. 7. Prediction results of CNN, CNN–LSTM, BWCL in single-cell minute.

Table 5
Single cell hourly prediction error performance table.

Data types	MAE	MSE	RMSE	R^2
Mean Low-frequency data	0.34719652	0.25254357	0.50253713	0.98696964
Mean Residual data	0.20238623	0.06626514	0.25742018	0.81214453
CNN Mean	1.407044	3.2667015	1.8074019	0.8360517
CNN–LSTM Mean	1.2424612	2.5459008	1.595588	0.872227
BWCL Mean	0.4070081	0.3216286	0.5671231	0.9824678
Peak Low-frequency data	0.5714517	0.47518077	0.68933356	0.97992595
Peak Residual data	0.31981915	0.1698277	0.41210157	0.81412760
CNN Peak	2.1784644	7.1043115	2.6653914	0.7155636
CNN–LSTM Peak	1.8988754	5.380438	2.4658544	0.7565565
BWCL Peak	0.6308704	0.5883823	0.7670608	0.9764428
Minimum Low-frequency data	0.34400505	0.22979534	0.47936973	0.98558828
Minimum Residual data	0.20951997	0.07294183	0.27007747	0.82533679
CNN Minimum	1.5522262	3.5602798	1.8868704	0.7847524
CNN–LSTM Minimum	1.2193955	2.327521	1.5898179	0.8471910
BWCL Minimum	0.4188073	0.3177259	0.5636718	0.9807507

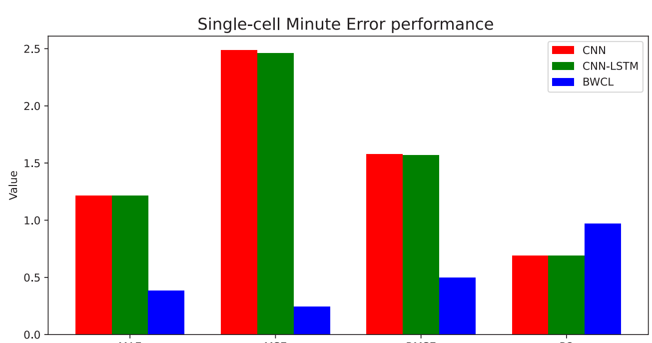

Fig. 8. Histogram of error performance for single-cell minute granularity.

Table 6
Multi-cell minute prediction error performance table.

Data types	MAE	MSE	RMSE	R^2
Low-frequency data	0.15800741	0.046740517	0.21619555	0.99335609
Residual data	0.30914626	0.16385198	0.40478635	0.80337199
CNN	1.4531637	3.3313746	1.8252053	0.5825370
CNN–LSTM	1.2126384	2.4796443	1.5746887	0.6892695
BWCL	0.3660354	0.2390515	0.4889290	0.9760884

accurate relative to a single cell. This can be attributed to the following two main reasons: First, in multi-cell environments, the transmission of mobile network traffic data has more complex spatial correlation and temporal dependence. CNN and CNN–LSTM models are not sufficient to effectively capture the spatio-temporal properties of time series. Secondly, there are more interfering factors in a multi-cell environment, which can affect the quality of network traffic data. Before making predictions, the data need to be processed appropriately, including data preprocessing steps such as noise filtering, signal cleaning, and interference suppression, to ensure the quality of the input data to improve the prediction performance of the model. BWCL prediction model proposed in this paper solves the above existing problems. As shown in Table 6, the prediction accuracy of the BWCL model is as high as 97%. Compared with CNN and CNN–LSTM models, its prediction accuracy is improved by 67.69% and 41.65%, respectively. The MAE, MSE, and $RMSE$ performance metrics were all reduced by 70.24%, 90.68%, and 69.42% accordingly. At the minute time granularity, multivariate improved the accuracy of the prediction results by 0.42% over univariate. Although the multi-cell prediction is slightly less computationally efficient (10 s), it fully takes into account the influence of neighboring cells on the target cell in time and space. This enables the system to respond more comprehensively to changes in traffic data trends, more accurately predict value of future traffic data, and then more rationally allocate network bandwidth to provide better network quality of service (see Fig. 15).

4.3.4. Analysis of multi-cell hourly prediction results

For multi-cells hourly time granularity, the minimum, mean, and peak values in each hour are extracted for prediction similarly. Compared with single cell, we construct more complex CNN, CNN–LSTM, and BWCL, and perform predictive analysis on network traffic data of these three different attributes. At the same time, we compare the error performance and computational efficiency of multi-cells hourly prediction and single-cell hourly prediction, taking into account the multi-perspective analysis.

As shown in Figs. 16–18, the experimental plots of the mean, minimum, and peak values of the multi-cells prediction in CNN, CNN–LSTM, and BWCL are shown. In Table 7, the R^2 of the mean, minimum, and peak values of the low-frequency components are 0.9861, 0.9870, and 0.9881, respectively, and the corresponding values of MAE, MSE, and $RMSE$ are also smaller. It can be seen that both the mean, minimum, and peak values of flow data are well predicted on low-frequency data.

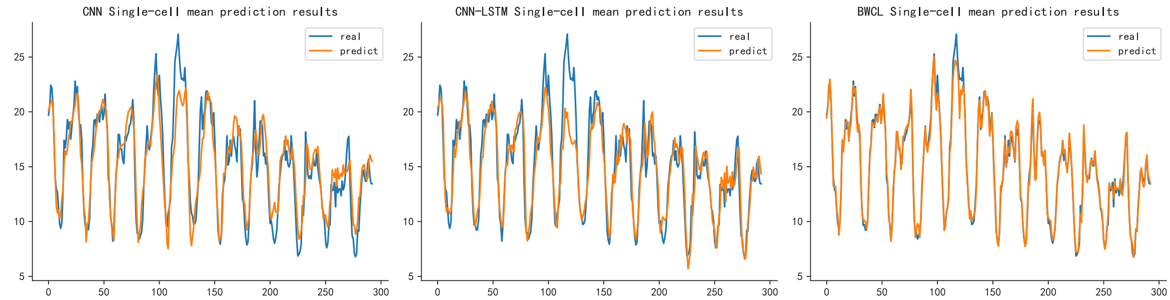

Fig. 9. Mean prediction results of CNN, CNN–LSTM, BWCL at single-cell hour granularity.

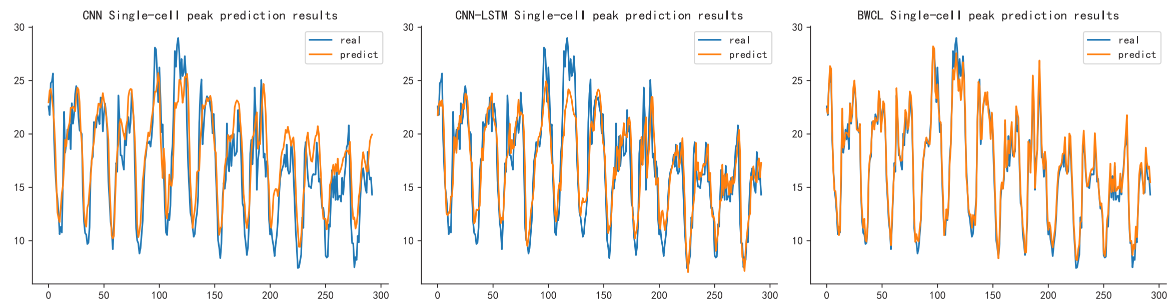

Fig. 10. Peak prediction results of CNN, CNN–LSTM, BWCL at single-cell hour granularity.

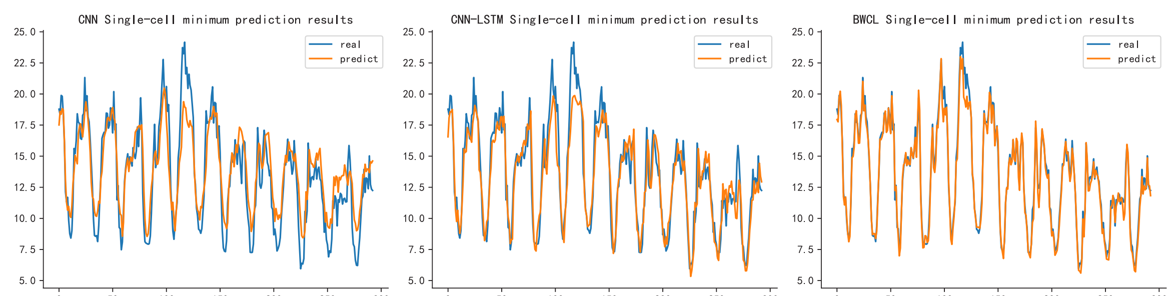

Fig. 11. Minimum prediction results of CNN, CNN–LSTM, BWCL at single-cell hour granularity.

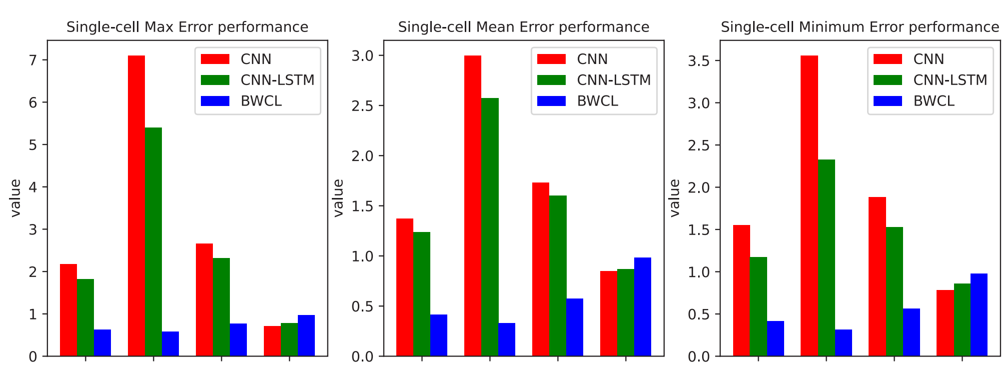

Fig. 12. Histogram of prediction error performance for single-cell hourly granularity.

In contrast, the residual component of network traffic data predicts slightly worse results due to the inhomogeneous distribution of the residual data, the nature of random oscillations, and the absence of autocorrelation. This makes it difficult to fully exploit the key feature information in the residual data, which makes its prediction performance slightly lower than that of the frequency signal. However, the R^2 of their prediction results all reach 0.80 and above accordingly. This also indicates that the prediction model proposed in this paper has strong generalization ability and can effectively capture its potential spatio-temporal characteristics. As can be seen from Table 7, compared with the prediction results of CNN and CNN–LSTM, the R^2 of the mean value is improved from 0.7776884 and 0.8174993 to 0.9827219, and its prediction accuracy is improved by 26.38% and 20.19%, respectively. The prediction accuracy of the peak is enhanced by 37.01% and 34.38%. The R^2 of the minimum is improved from 0.7877780 and 0.8711095 to 0.9834640, which is 24.84% and 12.89%, respectively. The corresponding MAE, MSE, and $RMSE$ are all reduced by more than 70%. It can be seen that the BWCL model has a high level of

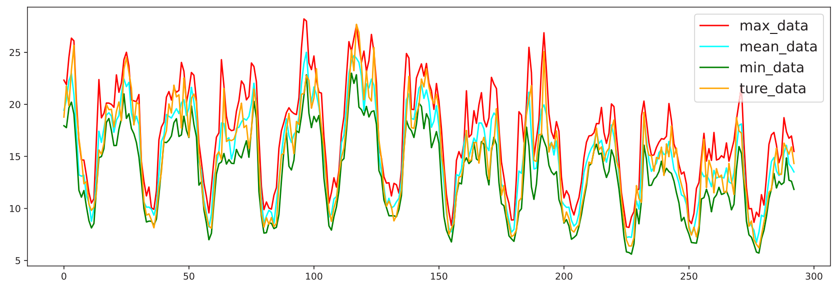

Fig. 13. Comparison between predicted and actual values of mean, peak, and minimum values in the single cell.

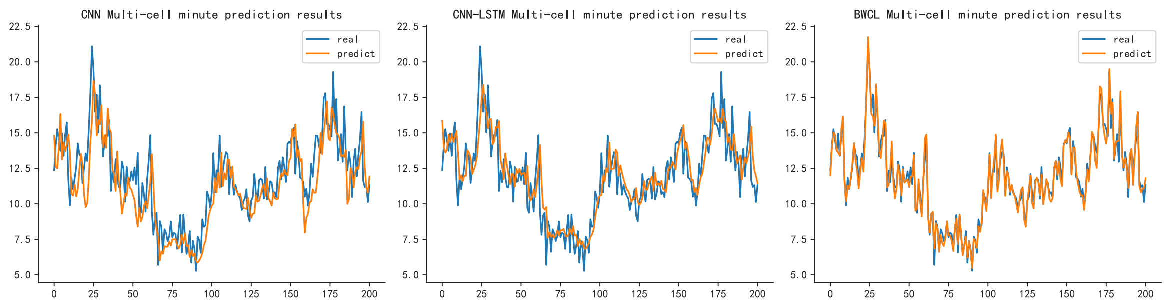

Fig. 14. Prediction results of CNN, CNN–LSTM, BWCL in Multi-cell minute.

Fig. 15. Histogram of prediction error performance for multi-cell minutes granularity.

prediction accuracy for mean, peak, and minimum values of network traffic data, which highlights the strong robustness of the model in prediction tasks (see Fig. 19).

Compared to the aforementioned single-cell prediction, this experiment shows a certain degree of improvement in prediction accuracy, but also slight decrease in computational efficiency. This is due to the complex spatio-temporal correlations among multiple cells. Therefore, the BWCL model needs to dig deeper into the spatial hidden features of time-series data and capture the long-term dependence of time more comprehensively to improve the prediction accuracy of network traffic data. Although the computing efficiency is slightly reduced, we consider more environmental factors in real-world scenarios. As a result, the multi-cells prediction results are more reliable and stable when analyzing real network traffic data.

Fig. 20 shows the complete prediction curves for the mean, minimum, and peak of the multi-cell hourly values and compares them to the real data. The trend of the predicted data can be observed very intuitively, and there is also a high prediction accuracy at the real data peaks. Multi-cells are more informative than single-cell prediction ranges. This is because multi-cells not only dig deeper into the potential characteristics of the data but also pay more attention to the impact of spatio-temporal correlation on future data.

5. Conclusion and future work

Accurately understanding and analyzing network traffic is critical for network planning, performance optimization, and security management. However, with the expansion of network scale and the introduction of new services. Network traffic shows diverse characteristics, such as mutations, trends, and randomness. This makes accurate traffic prediction a challenging task. To address this challenge, we propose the BWCL comprehensive prediction model for the first time. In the overall experiments, we focus on the comparison of CNN, CNN–LSTM, and BWCL models. The effect of different temporal granularities (minutes and hours) on the accuracy of single and multi-cell network traffic prediction is analyzed. At the minute time granularity, The $RMSE$ and R^2 of the BWCL model are both about 0.47 and 0.97 on both single and multi-cells. The prediction performance is improved by about 70% and 40% compared to unfiltered CNN–LSTM prediction results. This significant improvement is due to the application of the Butterworth filter. It effectively eliminates the high-frequency noise component, improves the data quality, and reduces the interference of noise on the low-frequency prediction model. The performance of the BWCL model is equally encouraging at the hourly time granularity. The $RMSE$ and R^2 of the overall prediction results are both improved by about 65% and 12% compared to the CNN–LSTM experimental results. This shows that Butterworth's application has a positive impact on the prediction of data of various temporal granularities and different statistical characteristics. In addition, we introduce residual data predictions, which usually highlight the fluctuations and irregularities of the data better and help the model capture the nonlinear and non-smooth features of network traffic more accurately. This enables us to improve the accuracy of the overall prediction of network traffic data while ensuring the integrity of network traffic data.

The future work plan focuses on two main areas. First, we will delve into more advanced filtering methods to reduce noise in network

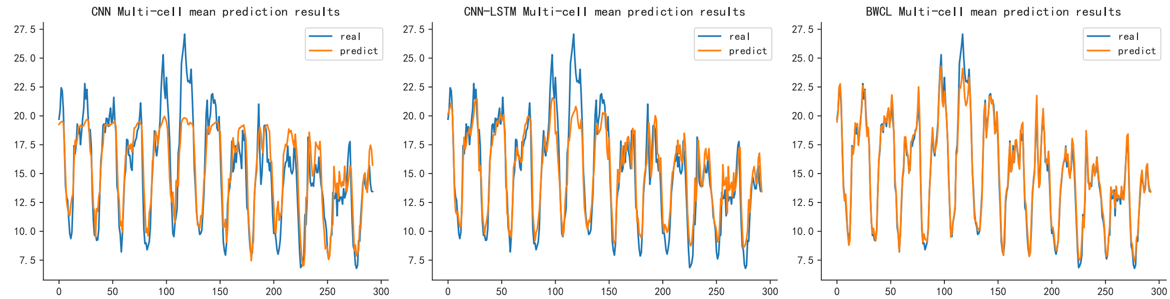

Fig. 16. Mean prediction results of CNN, CNN–LSTM, BWCL at multi-cell hour granularity.

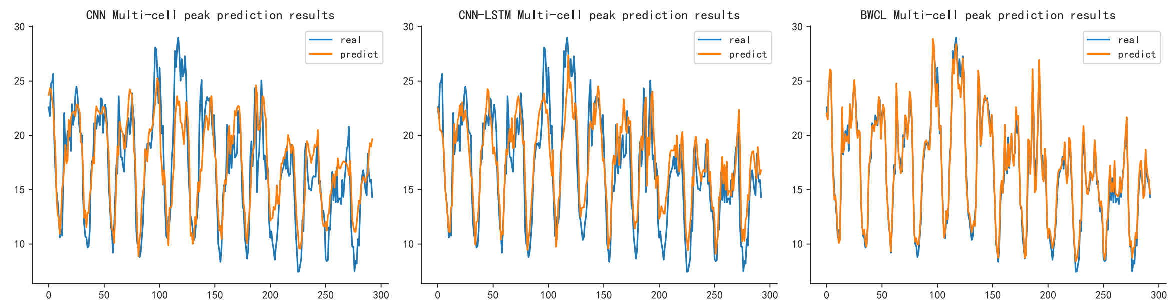

Fig. 17. Peak prediction results of CNN, CNN–LSTM, BWCL at multi-cell hour granularity.

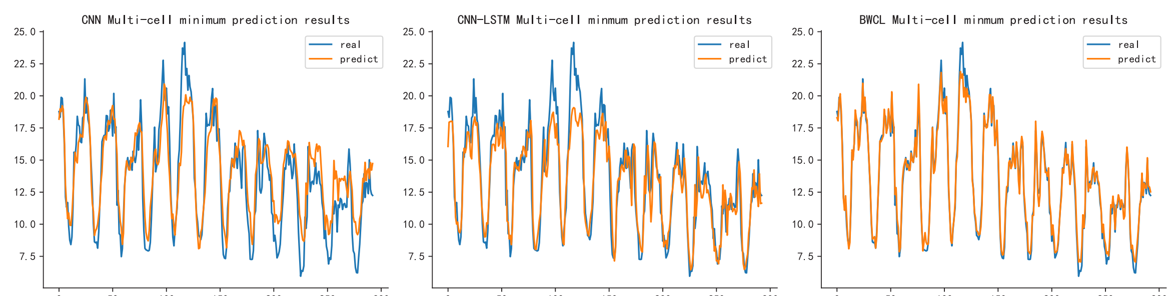

Fig. 18. Minimum prediction results of CNN, CNN–LSTM, BWCL at multi-cell hour granularity.

Table 7
Single cell hourly prediction error performance table.

Data types	MAE	MSE	RMSE	R^2
Mean Low-frequency data	0.40310398	0.26860094	0.51826730	0.98614114
Mean Residual data	0.17750953	0.050988138	0.22580554	0.85545342
CNN Mean	1.6297297	4.229601	2.1046617	0.7776884
CNN–LSTM Mean	1.4560539	3.0363616	1.7069246	0.8174993
BWCL Mean	0.4588987	0.3442691	0.5867445	0.9827219
Peak Low-frequency data	0.22745013	0.10109685	0.3179573	0.98813658
Peak Residual data	0.30572128	0.15720332	0.39648873	0.82794467
CNN Peak	2.1151419	7.083542	2.6614923	0.7163951
CNN–LSTM Peak	2.074484	6.7371354	2.5955992	0.7302642
BWCL Peak	0.5322696	0.4543997	0.6740918	0.9818071
Minimum Low-frequency data	0.31735855	0.20725931	0.45525742	0.98700164
Minimum Residual data	0.2124004	0.0707406	0.26597115	0.83060761
CNN Minimum	1.5282816	3.510235	1.8735621	0.7877780
CNN–LSTM Minimum	1.5080202	2.531899	1.6201024	0.8711095
BWCL Minimum	0.4012136	0.273511	0.47351141	0.9834640

traffic sequences. Second, we work on embedding the model into SDN based on the IoT-Fog networks. We will utilize the Butterworth filtering technique to smooth and preprocess the traffic data from different IoT devices to improve the data quality. We fully utilize the BWCL model to extract the key spatio-temporal features of the traffic data to improve the data processing efficiency of the SDN controller and fog computing. For sensitive data on mobile terminal devices, we will use differential privacy techniques or data desensitization methods to ensure that the privacy of individual data is protected. This series of work programs will lead to important breakthroughs in the areas of

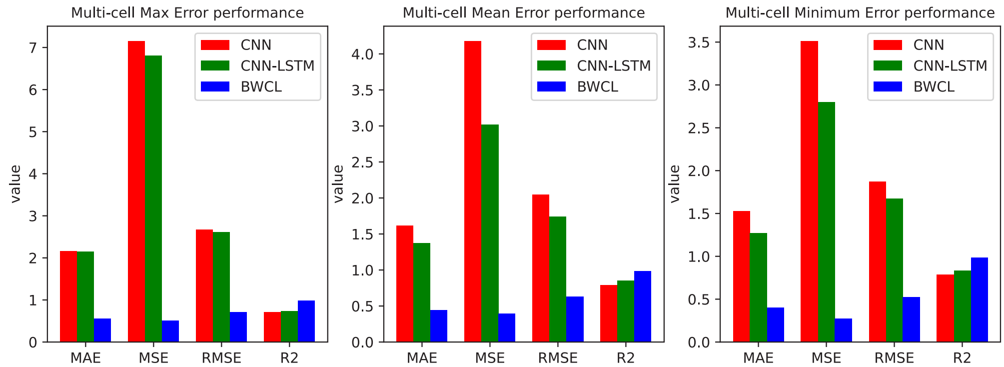

Fig. 19. Histogram of prediction error performance for multi-cell hourly granularity.

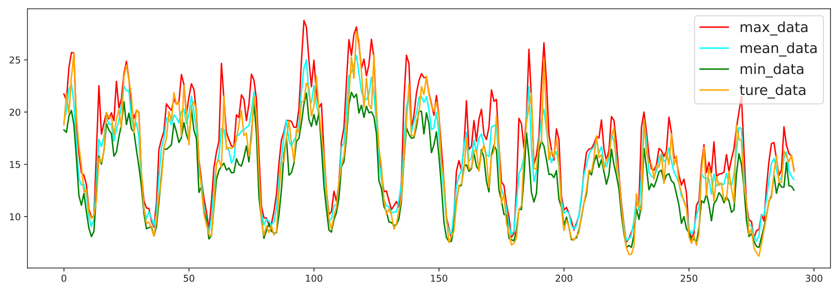

Fig. 20. Comparison of predicted and actual values of mean, peak, and minimum values for multiple plots.

network traffic analysis and IoT-Fog networks computing, promising to improve network performance and protect data privacy.

CRediT authorship contribution statement

Xueyan Hu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Wei Liu:** Writing – review & editing, Visualization, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Hua Huo:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research is financially supported by the Central Government Guiding Local Science and Technology Development Fund Program (No. Z20221343032), and the Major Science and Technology Program of Henan Province (No. 221100210500). All authors have read and agreed to the published version of the manuscript.

References

[1] Z. Gao, Y. Gu, 5G traffic prediction based on deep learning, in: Computational Intelligence and Neuroscience, 2022, pp. 1687–5265, http://dx.doi.org/10.1155/2022/3174530.

[2] X. Liang, Research on network security filtering model and key algorithms based on network abnormal traffic analysis, in: 2021 International Conference on Networking, Communications and Information Technology (NetCIT), 2021, pp. 226–229, http://dx.doi.org/10.1109/NetCIT54147.2021.00053.

[3] J. Wang, A process level network traffic prediction algorithm based on ARIMA model in smart substation, in: 2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013), 2013, pp. 1–5, http://dx.doi.org/10.1109/ICSPCC.2013.6663896.

[4] M. Cao, V.O.K. Li, V.W.S. Chan, A CNN-LSTM model for traffic speed prediction, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1–5, http://dx.doi.org/10.1109/VTC2020-Spring48590.2020.9129440.

[5] X. Zhang, S. Li, C. Zhang, Short-term prediction of SDN network traffic based on CEEMDAN and mixed kernel least squares support vector machine, in: 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), 2022, pp. 1201–1206, http://dx.doi.org/10.1109/PRAI55851.2022.9904179.

[6] M. Alkasassbeh, A novel hybrid method for network anomaly detection based on traffic prediction and change point detection, J. Comput. Sci. 14 (2) (2018) 153–162, http://dx.doi.org/10.3844/jcssp.2018.153.162.

[7] J. Kuang, D. Zhai, X. Wu, Y. Wang, Q. Ma, A network traffic prediction method using two-dimensional correlation and Single exponential smoothing, in: 2013 15th IEEE International Conference on Communication Technology, 2013, pp. 403–406, http://dx.doi.org/10.1109/ICCT.2013.6820409.

[8] B.-S. Chen, S.-C. Peng, K.-C. Wang, Traffic modeling, prediction, and congestion control for high-speed networks: a fuzzy AR approach, IEEE Trans. Fuzzy Syst. 8 (5) (2000) 491–508, http://dx.doi.org/10.1109/91.873574.

[9] E. Erdem, J. Shi, ARMA based approaches for forecasting the tuple of wind speed and direction, Appl. Energy 88 (4) (2011) 1405–1414, http://dx.doi.org/10.1016/j.apenergy.2010.10.031.

[10] Y. Yu, J. Wang, M. Song, J. Song, Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient, in: 2010 International Conference on Intelligent System Design and Engineering Application, Vol. 1, 2010, pp. 980–983, http://dx.doi.org/10.1109/ISDEA.2010.335.

[11] L. Ye, S. Ding, C. Liu, W. Zhu, Network traffic prediction based on double-synchronized periodic LSTM, in: 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), 2022, pp. 409–410, http://dx.doi.org/10.1109/COMPSAC54236.2022.00065.

[12] J. Guo, Z. Xie, Y. Qin, L. Jia, Y. Wang, Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM, IEEE Access 7 (2019) 42946–42955, http://dx.doi.org/10.1109/ACCESS.2019.2907739.

[13] A. Thaduri, V. Polepally, S. Vodithala, Traffic accident prediction based on CNN model, in: 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1590–1594, http://dx.doi.org/10.1109/ICICCS51141.2021.9432224.
[14] S. Nihale, S. Sharma, L. Parashar, U. Singh, Network traffic prediction using long short-term memory, in: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 338–343, http://dx.doi.org/10.1109/ICESC48915.2020.9156045.
[15] I.I. Turulin, H.S. Mogheer, Method and algorithm for synthesis of controlled digital low-pass butterworth filters on the example of a 4th order filter, in: 2022 International Ural Conference on Electrical Power Engineering (UralCon), 2022, pp. 336–340, http://dx.doi.org/10.1109/UralCon54942.2022.9906684.
[16] N. Ramakrishnan, T. Soni, Network traffic prediction using recurrent neural networks, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 187–193, http://dx.doi.org/10.1109/ICMLA.2018.00035.
[17] C. Zhang, H. Zhang, J. Qiao, D. Yuan, M. Zhang, Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1389–1401, http://dx.doi.org/10.1109/JSAC.2019.2904363.
[18] Q.T. Tran, L. Hao, Q.K. Trinh, Cellular network traffic prediction using exponential smoothing methods, J. Inf. Commun. Technol. 18 (1) (2018) 1–18, http://dx.doi.org/10.32890/jict2019.18.1.1.
[19] C. Bhar, N. Chatur, A. Mukhopadhyay, G. Das, D. Datta, Designing a green optical network unit using ARMA-based traffic prediction, in: 2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), 2015, pp. 1–6, http://dx.doi.org/10.1109/ANTS.2015.7413604.
[20] J. Lee, H. Lee, N. Kim, B. Lee, A study on online ARIMA algorithms applying various gradient descent optimization algorithms for time series prediction, in: 2021 International Conference on Information and Communication Technology Convergence (ICTC), 2021, pp. 1104–1106, http://dx.doi.org/10.1109/ICTC52510.2021.9620990.
[21] L. Deng, K. Ruan, X. Chen, X. Huang, Y. Zhu, W. Yu, An IP network traffic prediction method based on ARIMA and N-BEATS, in: 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS), 2022, pp. 336–341, http://dx.doi.org/10.1109/ICPICS55264.2022.9873564.
[22] S. Zhang, M. Han, M. Xu, Chaotic time series online prediction based on improved kernel adaptive filter, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–6, http://dx.doi.org/10.1109/IJCNN.2018.8489753.
[23] X. Jiang, L. Zhang, X. (Michael) Chen, Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China, Transp. Res. C 44 (2014) 110–127, http://dx.doi.org/10.1016/j.trc.2014.03.016.
[24] L. Nie, Z. Ning, M.S. Obaidat, B. Sadoun, H. Wang, S. Li, L. Guo, G. Wang, A reinforcement learning-based network traffic prediction mechanism in intelligent internet of things, IEEE Trans. Ind. Inform. 17 (3) (2021) 2169–2180, http://dx.doi.org/10.1109/TII.2020.3004232.
[25] W. Liang, Y. Li, K. Xie, D. Zhang, K.-C. Li, A. Souri, K. Li, Spatial-temporal aware inductive graph neural network for C-ITS data recovery, IEEE Trans. Intell. Transp. Syst. 24 (8) (2023) 8431–8442, http://dx.doi.org/10.1109/TITS.2022.3156266.
[26] J. Gaboitaolelwe, A.M. Zungeru, A. Yahya, C.K. Lebekwe, D.N. Vinod, A.O. Salau, Machine learning based solar photovoltaic power forecasting: A review and comparison, IEEE Access 11 (2023) 40820–40845, http://dx.doi.org/10.1109/ACCESS.2023.3270041.
[27] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, C. Peng, Spatio-temporal analysis and prediction of cellular traffic in metropolis, IEEE Trans. Mob. Comput. 18 (9) (2019) 2190–2202, http://dx.doi.org/10.1109/TMC.2018.2870135.
[28] X. Xu, S. Gao, Z. Jiang, LSTCN: An attention-based deep neural network model combining LSTM and TCN for cellular network traffic prediction, in: 2021 5th International Conference on Communication and Information Systems (ICCIS), 2021, pp. 34–38, http://dx.doi.org/10.1109/ICCIS53528.2021.9645961.
[29] N. Ranjan, S. Bhandari, H.P. Zhao, H. Kim, P. Khan, City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN, IEEE Access 8 (2020) 81606–81620, http://dx.doi.org/10.1109/ACCESS.2020.2991462.
[30] J. Fan, D. Mu, Y. Liu, Research on network traffic prediction model based on neural network, in: 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), 2019, pp. 554–557, http://dx.doi.org/10.1109/ICISCAE48440.2019.221694.
[31] Z. Zhao, Z. Li, F. Li, Y. Liu, CNN-LSTM based traffic prediction using spatial-temporal features, J. Phys. Conf. Ser. 2037 (1) (2021) 012065, http://dx.doi.org/10.1088/1742-6596/2037/1/012065.
[32] H. Shi, C. Pan, L. Yang, X. Gu, AGG: a novel intelligent network traffic prediction method based on joint attention and GCN-GRU, Secur. Commun. Netw. 2021 (2021) 7751484:1–7751484:11, http://dx.doi.org/10.1155/2021/7751484.
[33] Y. Peng, T. Feng, C. Yang, C. Leng, L. Jiao, X. Zhu, L. Cao, R. Li, HMM-LSTM for proactive traffic prediction in 6G wireless networks, in: 2021 IEEE 21st International Conference on Communication Technology (ICCT), 2021, pp. 544–548, http://dx.doi.org/10.1109/ICCT52962.2021.9657967.
[34] Y. He, Y. Yang, B. Zhao, Z. Gao, L. Rui, Network traffic prediction method based on multi-channel spatial-temporal graph convolutional networks, in: 2022 IEEE 14th International Conference on Advanced Infocomm Technology (ICAIT), 2022, pp. 25–30, http://dx.doi.org/10.1109/ICAIT56197.2022.9862813.
[35] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, 2020, http://dx.doi.org/10.1609/aaai.v35i12.17325, ArXiv abs/2012.07436.

Xueyan Hu received the B.S. degrees in Zhengzhou Business University, Zhengzhou, China, in 2021. She is currently pursuing the M.S. degree in Henan University of Science and Technology, China. Her current research interests include mobile network traffic prediction and anomaly detection, deep learning for signal detection and decoding.

Wei Liu received the bachelor of science degree from Lanzhou University, China, in 1990 and the Master of Engineering degree from the Xidian University, China, in 2000, and Ph.D. degree from the same university, in 2000. He is a professor in the Henan University of Science and Technology. His research interests include wireless communication, machine learning and data networks.

Hua Huo received the Ph.D. degree in computer science and technology from the Xi'an Jiaotong University in China, in 2005. He was a postdoctoral research fellow at Tsinghua University in China, from 2006 to 2008, and at Trinity College Dublin in Ireland, from 2009 to 2010. He has been a professor at Henan University of Science and Technology in China, since 2012. His current research focuses on Intelligent Information Processing and Intelligent Computation for BigData.