Wireless Network Traffic Prediction in Cellular Communication Networks

Liangzhi Wang

Department of Electronic and Electrical Engineering

University of Sheffield

Supervisors: Prof. Jie Zhang, Prof. Xiaoli Chu

This thesis is submitted for the approval of the

Doctor of Philosophy

March 2024

This thesis is dedicated to my beloved family and friends. Without their unconditional love, encouragement and support, I would not be the person I am today.

Acknowledgements

I would like to express my sincerest gratitude to my first supervisor, Prof. Jie Zhang, for all he has done for me during my Ph.D studies. He consistently offers forward-thinking insights into my research direction, provides highly valuable suggestions for my work, and demonstrates remarkable patience and adaptability. Moreover, beyond academia, his optimistic and open-minded approach to life has profoundly impacted me. I would also like to express my appreciation to my second supervisor, Prof. Xiaoli Chu, for her guidance since my Ph.D studies. She consistently goes above and beyond to offer me selfless assistance.

Special thanks are extended to Prof. Jiliang Zhang, Dr. Zitian Zhang and Dr. Chen Chen, whose significant contributions have facilitated my transformation from a novice to an autonomous researcher. My Ph.D studies would have been unattainable without their unwavering guidance and assistance.

I would also like to extend my appreciation to Dr. Fuyou Li, Dr. Bo Ma, Prof. Haonan Hu, Dr. Ju Tan, and Dr. Jixuan Lin for their collaboration, insights, and assistance in various aspects of this research. Specifically, I am very thankful to Dr. Yu Yao, Mr. Wenji Xi and Mr. Chenyang Yuan, who gave me full support and went through thick and thin together with me. I would also like to thank Mr. Yangyang Xue, Mr. Songyan Zhang, Mr. Sipei Wang, Mr. Haoyuan Zhu, Mr. Lingyou Zhou, Mr. Jinbo Hou, Mr. Gang Yu, Mr. Jiaqi Liu and Mr. Xin Dong. Their friendship helps me a lot.

Last but not the least, I want to thank my family, my wife Zimiao Xu, my parents Prof. Hongliang Wang and Ms. Bingli Wang, my parents-in-law Mr. Dong Xu and Ms. Lei Guo, whose encouragement and understanding have been a constant source of motivation throughout my Ph.D studies. I am truly grateful for their unwavering support.

Abstract

Wireless network traffic prediction (NTP) is regarded as one of the most significant techniques for alleviating network resource pressure. However, existing methods struggle to balance the prediction accuracy, interpretability, and computational efficiency when dealing with aggregate-level wireless network traffic data. Additionally, existing NTP methods are inadequate in effectively addressing the nonroutine traffic caused by nonroutine events. Furthermore, there is a lack of specialized hyper-parameter optimization method for deep learning models when applied to cell-level wireless network traffic data. To address these challenges, this thesis proposes three key contributions, each corresponding to one of the aforementioned motivations. While all of these contributions focus on the domain of NTP, each targets a different application scenario and collectively addresses specific gaps in the field from various perspectives.

In **Work 1**, a novel user-behavior-based (UBB) NTP method is proposed for aggregate-level wireless network traffic data. Based on the analysis of overall user behavior, we utilize three traffic components to construct the daily NTP model with high interpretability. In addition, the initial parameter selection strategy of the UBB NTP method is discussed. The numerical results indicate the method has a high level of computational efficiency and prediction accuracy in comparison with traditional statistics-based and machine learning-based methods.

Work 2 addresses the special case of aggregate-level network traffic data, focusing on traffic that has been influenced by nonroutine events. Specifically, some nonroutine events have a strong impact on user behavior and then trigger the nonroutine network traffic. Therefore, we propose an innovative nonroutine network traffic prediction (NNTP) method

and then propose the soccer game (SG)-NNTP model as a case study in both single-step and multi-step prediction modes. Experimental results indicate the NNTP method is well-suited to this scenario, and far superior to benchmark methods in terms of interpretability, prediction accuracy, and computational efficiency.

In **Work 3**, we propose a meta-learning based framework for optimizing hyper-parameters in deep neural network-based NTP models when processing cell-level wireless network traffic data. The cell-level wireless network traffic data tends to exhibit a high degree of complexity due to the limited coverage, number of users, and mobility of users. Therefore, it poses a high demand on the learning capacity of NTP models. We propose an attention based deep neural network (ADNN) for the cell-level wireless NTP, namely the base-learner, in **Work 3**. More importantly, we propose a meta-learning based hyper-parameter optimization framework, i.e., the meta-learner. It can automatically provide proper hyper-parameters to match newly-given base-learners. Experimental results demonstrate the innovative meta-learner can further enhance the potential of the base-learner, and is robust for other deep learning-based models.

List of Publications

[1]. **L. Wang**, J. Zhang, Z. Zhang and J. Zhang, “Analytic network traffic prediction based on user behavior modeling,” in *IEEE Networking Lett.*, vol. 5, no. 4, pp. 208-212, Dec. 2023.

[2]. **L. Wang**, H. Zhu, J. Zhang, Z. Zhang and J. Zhang, “Interpretable nonroutine network traffic prediction with a case study,” in *IEEE Trans. Green Commun. Networking*, (Under Review).

[3]. **L. Wang**, J. Zhang, Y. Gao, J. Zhang, G. Wei, H. Zhou, B. Zhuge and Z. Zhang, “Hyper-parameter optimization for cell-level wireless network traffic prediction with a novel meta-learning framework,” in *IEEE Internet Things J.*, (Under Review).

[4]. **L. Wang**, C. Chen, C. Fischione, and J. Zhang, “Learning-based joint antenna selection and precoding design for cell-free MIMO networks,” in *IEEE Trans. Commun.*, (Under Review).

Table of contents

List of figures

List of tables

Abbreviations

4G Fourth Generation

5G Fifth Generation

ADNN Attention-based Deep Neural Network

AGA Advanced Genetic Algorithm

AIC Akaike Information Criterion

AR Autoregressive

ARIMA Autoregressive Integrated Moving Average

ARMA Autoregressive Moving Average

BIC Bayesian Information Criterion

CNN Convolutional Neural Network

DBNG Deep Belief Network and Gaussian

DNN Deep Neural Network

EB Exabyte

ES Exhaustive Searching

ESN Echo-State Network

FFNN Feed-Forward Neural Network

GA Genetic Algorithm

GARCH Generalized Auto-Regressive Conditional Heteroskedasticity

GLU Gated Linear Units

GNN Graph Neural Network

GP Gaussian Process

GRN Gated Residual Network

GRU Gate Recurrent Unit

GSM Grid Spectral Mixture

GSMA Global System for Mobile Communications Association

IoE Internet of Everything

KNN K-Nearest Neighbor

LR Linear Regression

LSTM Long Short-Term Memory

MA Moving Average

MAE Mean Absolute Error

MIMO Multiple Input Multiple Output

ML Machine Learning

MLP Multilayer Perceptron

MSE Mean Square Error

NNTP Nonroutine Network Traffic Prediction

NTMA Network Traffic Monitoring and Analysis

NTP Network Traffic Prediction

PSO Particle Swarm Optimization

QoS Quality of Service

R2 Coefficient of Determination

ReLU Rectified Linear Unit

RF Random Forest

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

SARIMA Seasonal Autoregressive Integrated Moving Average

SG Soccer Game

SG-NNTP Soccer Game Nonroutine Network Traffic Prediction

SMS Short Message Service

SVM Support Vector Machine

SVR Support Vector Regression

Tanh Hyperbolic Tangent

UBB User Behavior Based

VR Virtual Reality

Chapter 1

Introduction

1.1 Background

Over the past few decades, the number of mobile subscribers and networked devices, along with the network traffic usage per device, has grown explosively. By the end of the third quarter of 2023, the global wireless network traffic data reached an impressive 143 exabyte (EB) [1]. Therefore, network resource management faces severe challenges.

Since the official commercialization and global deployment of fifth generation (5G) mobile communication system in 2019, 5G mobile subscriptions have increased rapidly. According to the statistics of Ericsson, by 2023, the number of global mobile subscriptions has reached 8.5 billion, of which 1.4 billion are 5G subscriptions [1]. Both Ericsson and Global System for Mobile Communications Association (GSMA) forecast that the 5G technology will overtake the fourth generation (4G) technology, and become the dominant mobile technology in 2029 [1][2]. GSMA predicts that the 5G subscriptions will take up 54% of all subscriptions, approximately 5.29 billion in 2030 [2], while Ericsson forecasts that it will surpass 5.3 billion by 2029. According to Ericsson's report [1], the 5G mobile subscriptions are predicted to increase by a factor of 3.79, and take up 58% of all subscriptions, as shown in Fig. 1.1.

Moreover, numerous new networking paradigms have emerged, such as millimeter wave communication, massive Multiple Input Multiple Output (MIMO) network, ultra-dense network, promoting the rapid development and deployment of numerous technologies such as virtual reality (VR) and Internet of everything (IoE) [3]. These technologies will generate extremely large amounts of wireless network traffic while providing users with high-quality network services. Following Ericsson's forecast, the global monthly wireless network traffic data will reach 56 G per smartphone by the end of 2029 [1]. In addition to smartphones, various other network devices, such as smart home devices, smart city settings, etc., also contribute to the huge volume of global wireless network traffic. As per Ericsson's report, the number of cellular IoT connections will reach 6.1 billion and the global wireless network traffic data will reach around 563 EB per month by 2029 [1].

Such a huge volume of wireless network traffic data poses severe challenges to network resource management. To address these challenges, wireless network traffic prediction (NTP) has gained considerable attention from both industry and academia [4] [5] [6].

1.2 NTP Technology and Evolution

1.2.1 NTP Technology

NTP technology plays an important role in network traffic monitoring and analysis (NTMA) [6] [7] [8]. By analyzing and learning historical wireless network traffic data, NTP models are able to extract the traffic patterns and then precisely predict future traffic demands. NTP is an essential foundation of resource provisioning and congestion control. For instance, based on future traffic load prediction, an optimal sleeping strategy can be implemented for multiple cooperative access points to reduce their energy consumption [9]. In addition, many of the emerging intelligent architectures for cellular networks [10] rely on accurate cell-level wireless network traffic prediction to further enhance network performance.

NTP performance encompass accuracy, computational efficiency, and interpretability. Accurate traffic prediction enables more precise alignment with resource requirements, thereby preventing both over-allocation and under-allocation of resources. Computational efficiency is crucial for reducing computation time and resource consumption, ensuring timely responses and rapid resource adjustments. In the context of network management, interpretable

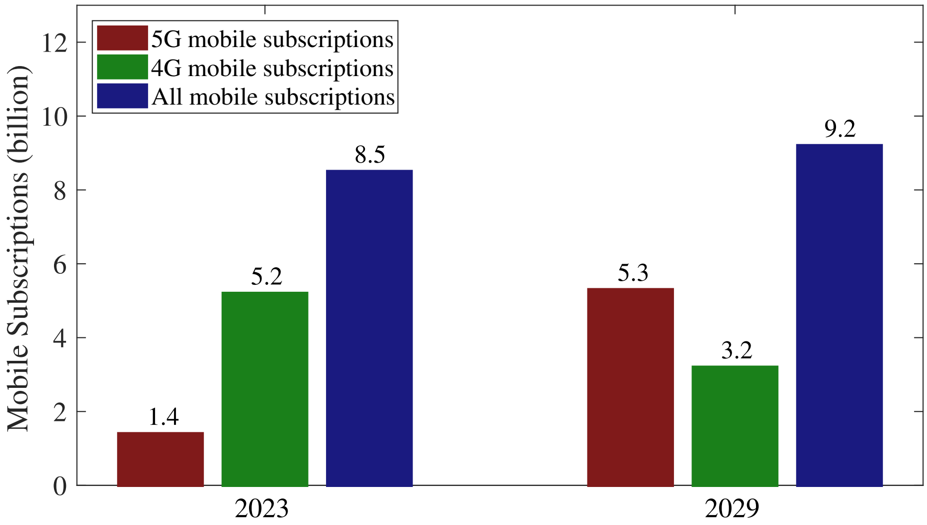


Fig. 1.1 Mobile subscriptions in 2023 and 2029 predicted by Ericsson.

prediction models provide operators with a clearer understanding of the underlying factors driving the predicted outcomes, facilitating more informed and rational decision-making.

NTP has been widely modeled as a time-series forecasting problem, in which wireless network traffic data points are arranged in chronological order and future wireless network traffic is inferred based on the correlation among data points in historical data. NTP usually consists of the following steps:

- Data Acquiring and pre-processing: Collect historical wireless network traffic data and process missing values and outliers.In addition, some works perform the operations such as decomposition and clustering on the traffic data [4].

- Model Selection: Select the proper models for NTP tasks, such as the traditional statistics-based methods and machine learning (ML)-based methods.

- Data Fitting: Adjust model parameters so that the NTP model can fit historical wireless network traffic data well.

- Prediction: Execute the NTP model and predict future traffic.

- Model Evaluation: Evaluate the performance of NTP models in terms of prediction accuracy, computational efficiency, and interpretability. Performance metrics for prediction accuracy include Mean Square Error (MSE), coefficient of determination (R2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Elapsed time of a NTP model can be adopted as the performance metric for computational efficiency.

However, to some extent, there is a certain conflict between the prediction accuracy and interpretability of NTP. Specifically, simple NTP models usually possess high interpretability and computational efficiency; however, their prediction accuracy is usually poor. On the contrary, the complicated NTP models such as deep learning-based models, are designed for higher prediction accuracy at the expense of interpretability and computational efficiency [11]. It is worth noting that the more complex a neural network becomes, the less interpretable it is, often turning into a "black box", where understanding how decisions are made becomes increasingly difficult. Meanwhile, complicated NTP models introduce some other tricky problems like hyper-parameter selection.

1.2.2 Evolution of NTP Technology

NTP technology has experienced an evolution from traditional statistics-based methods to machine-learning based methods. Traditional statistics-based methods are built on mathematical models or probability distributions with simple structure and relatively fewer parameters. Moreover, these models generally rely on a priori assumptions, such as a certain distribution of data, and there are certain correlation among variables, which makes the model clearer and easier to understand. Most of these models are linear models, such as the Autoregressive Moving Average (ARMA) model and the Autoregressive Integrated Moving Average (ARIMA) model. Very few of these models focus on nonlinear characters of time-series data such as the Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) model. The distinctive features of traditional statistics-based methods are that these models are simple with small number of parameters. These features bring the advantages of high interpretability and computational efficiency, but at the cost of prediction accuracy.

With the exponential growth of wireless network traffic data and the rapid development of ML technology, the data-driven ML-based NTP models emerge constantly and exhibit the superiority in prediction accuracy. The application of multiple activation functions provides ML-based NTP models with the capacity to understand complex nonlinear relationships. ML-based NTP models include two categories, namely shallow learning-based NTP models and deep learning-based NTP models [12][13]. Shallow learning-based methods were introduced into NTP field earlier, and achieves improvement in prediction accuracy. However, these models are relatively simple and can hardly predict wireless network traffic data accurately, as their limited learning abilities may hinder the effective utilization of ample training data.

In the past few years, the deep learning in NTP filed has gained much attention. Deep learning-based NTP models empowered by deep neural networks (DNNs) perform well in terms of prediction accuracy. Equipped with multiple layers of neural networks, DNNs are able to capture the correlation in the wireless traffic data. On the other hand, each layer of the hierarchical structure performs non-linear transform of the input data, which makes the mapping relation more blurry. The complicated structure and non-linear transform enhance the prediction accuracy, in the meantime sacrifice the interpretability as well as computational efficiency. In addition, deep learning-based models involve numerous hyper-parameters such as the learning rate, the number of neural layers, the number of neurons in each layer, etc. In general, the deep learning-based methods have now become the dominant research direction in NTP.

1.3 Motivation and Contribution

1.3.1 Motivation

Recently, NTP has been extensively studied in the academia. However, there are still some open questions, such as the balance among interpretability, accuracy as well as computational complexity, the inspiration of user behavior, the impact of nonroutine events, and the hyper-parameter selection of deep learning-based NTP models. The motivations of this thesis are summarized as follows.

- As mentioned above, the existing works on NTP field fail to achieve a balance among interpretability, prediction accuracy, and computational complexity. Traditional statistics-based methods perform well in interpretability but poorly in prediction accuracy. In contrast, deep learning-based methods have high prediction accuracy but low interpretability.

- In the NTP field, the existing works focus only on historical wireless network traffic data, but have not taken into account user behavior [11]. These works have primarily applied traditional statistics-based or machine learning (ML)-based time series forecasting models to wireless NTP tasks, without fully considering and utilizing the user behavior characteristics that dominate wireless network traffic patterns. The NTP model constructed on the basis of user behavior can enhance the fitness of the model to the specific data, i.e., wireless network traffic data, and improve the model's performance including prediction accuracy, computational efficiency, as well as interpretability.

- In users' daily lives, certain events that affect user behavior will subsequently trigger nonroutine traffic, which will severely limit the performance of the NTP model. The nonroutine traffic is less frequent and does not have a regular recurring cycle. Thereby, it is difficult to extract accurate traffic patterns from only historical traffic data, both for traditional statistics-based and ML-based methods.

- It is hard to extract local users' overall behavior from the cell-level wireless network traffic data. This is because a mobile cell serves limited users with a small coverage area, and the users in the coverage area have strong mobility. Therefore, user behavior analysis cannot be utilized to construct NTP models that simultaneously enhance prediction accuracy, interpretability, and computational efficiency in this context. Among the remaining approaches, deep learning-based models demonstrate the highest prediction accuracy. Therefore, adopting deep learning techniques is the most suitable option to ensure superior prediction accuracy. Commonly used deep neural networks in NTP field include multilayer perceptron (MLP) network, recurrent neural network

(RNN), long short-term memory (LSTM) network, etc. RNN suffers from the vanishing/exploding gradient issue [14]. While LSTM addresses the vanishing/exploding gradient issue, its sequential processing mechanism hinders its performance. Thus, a novel deep learning-based NTP model is required.

- In addition, deep learning-based methods face the challenge of hyper-parameter selection. Deep learning-based cell-level wireless NTP models involve numerous hyper-parameters such as the learning rate, the number of neural layers, the number of neurons in each layer, etc. Unfortunately, how to efficiently optimize the hyper-parameters has not been well studied and is still an open question to the best of our knowledge. Furthermore, the access points in the 5G or beyond wireless networks are ultra-densely deployed and there are tens of thousands of mobile cells in large-scale radio access networks [15][16]. Hence, it is impractical to optimize the hyper-parameters for each mobile cell manually, and an efficient is required.

1.3.2 Contribution

To address the aforementioned challenges encountered in NTP filed, this thesis focuses on improving NTP models in three aspects including prediction accuracy, computational efficiency, and interpretability. Then, we summarize the main contributions of this thesis as follows.

- First, this thesis classifies traffic data into aggregate-level and cell-level categories and focuses on designing or enhancing different NTP methods tailored to the unique characteristics of each data type to improve prediction performance. The key distinction between these two data types lies in their inherent characteristics: cell-level traffic, generated by a single cell, involves a small geographic area and a limited number of users, leading to traffic patterns that exhibit significant randomness and frequent fluctuations. In contrast, aggregate-level traffic is generated by multiple neighboring cells, covering a larger area and serving a greater number of users, which results in more regular traffic patterns and smoother curves.

- Second, for aggregate-level wireless network traffic data, this thesis incorporates user behavior analysis into the construction of the NTP model and proposes a novel user behavior-based (UBB) NTP method. The structure of the UBB NTP model is clearer and easier to understand compared with existing works. The parameters of the proposed model are concise and have corresponding physical meanings, thus greatly improving the interpretability. The traditional statistics-based and deep learning-based methods are considered as benchmark schemes. Experiment results indicate that the proposed UBB NTP method outperforms benchmark methods in terms of overall performance. More specifically, the efficiency of the proposed method is approximately 12 times that of the ARMA model and 28 times that of the LSTM network.

- Third, this thesis takes the lead in defining and systematically analyzing the nonroutine traffic, and evaluates its impact on the benchmark methods in terms of prediction accuracy, computational efficiency and interpretability. Based on the observation that nonroutine traffic is closely related to the corresponding nonroutine event, this thesis proposes a novel nonroutine network traffic prediction (NNTP) method which takes full advantage of the nonroutine event's information to construct the NTP model with high interpretability. In addition, taking the real-world event as a case study, this thesis builds the soccer game (SG)-NNTP model following the NNTP method and tests both single-step and multi-step prediction mode. In the multi-step mode, the NTP model directly predict the traffic values for the next m (step size) moments, while the single-step one only focuses on the next one moment. In comparison with benchmark methods, the proposed NNTP method achieves the best prediction accuracy and computation efficiency in both single-step and multi-step prediction modes.

- Finally, this thesis proposes a novel attention based deep neural network for cell-level wireless NTP tasks. More importantly, to solve the common challenge of deep learning-based NTP models, i.e., hyper-parameter optimization, this thesis develops an innovative meta-learning based framework that selects the optimal hyper-parameters for cell-level NTP tasks automatically by analyzing and utilizing the correlations between the target NTP tasks and meta-samples. Finally, extensive simulations demonstrate that the proposed meta-learning based framework is more effective compared to the traditional hyper-parameter optimization methods and is robust across different base learners with various deep learning algorithms.

1.4 Structure of the Thesis

The content of this thesis is organized as follows.

Chapter 2: Literature Review

This chapter is a comprehensive review of the existing works on NTP field. We first review the traditional statistics-based NTP model and introduce two typical statistics-based methods, i.e. ARMA model and ARIMA model. Subsequently, the basic concepts and theories of ML are introduced. The widely used ML-based models are described, such as the MLP network and the LSTM network. Finally, this chapter reviews the state of the art deep learning-based NTP models and their merits and shortcomings.

Chapter 3: Analytic Network Traffic Prediction Based on User Behavior Modeling

This chapter analyzes the overall user behavior patterns of a certain region, and studies the relationship between the user behavior patterns and local wireless network traffic patterns. An UBB NTP method is proposed and the mathematical model is formulated. Finally, we evaluate the proposed UBB NTP method in comparison with ARMA model, ARIMA model, MLP network, and LSTM network.

Chapter 4: Nonroutine Network Traffic Prediction with A Case Study

This chapter focuses on nonroutine network traffic and analyzes the flaws of the existing NTP models in the face of nonroutine network traffic. Then, utilizing the correlations between the nonroutine network traffic and the nonroutine event, we propose a novel NNTP method with high interpretability, computational efficiency, and prediction accuracy. Taking the real-world soccer-games as a case study, we construct a SG-NNTP model with both multi-step and single-step prediction modes. Simulation results show that the NNTP method fits well with the nonroutine traffic data in both multi-step and single-step prediction modes.

Chapter 5: Hyper-parameter Optimization for Cell-level Wireless Network Traffic Prediction with A Novel Meta-Learning Framework

In the case where the overall user behavior is difficult to obtain, this chapter designs an attention based deep neural network for cell-level wireless NTP task. Furthermore, considering hyper-parameter selection problem, this chapter proposes an innovative meta-learning based hyper-parameter optimization framework. Finally, the performance and robustness of the proposed framework is validated by simulations.

Chapter 6: Conclusion and Future Work

This chapter summaries the thesis and gives the prospect of the future works.

Chapter 2

Literature Review

Overview

This chapter consists of two sections. The first one introduces statistics-based and deep learning-based NTP models, respectively, which are commonly used in existing works and also serve as the benchmark models in this thesis. We start from the statistics-based NTP models including the ARMA and ARIMA models. Then, we clarify some key concepts of deep learning and introduce two typical models in detail: the MLP and LSTM networks. In the second section, we conduct a comprehensive review of the literature on traditional statistics-based models, shallow learning-based models, and deep learning-based models, respectively, and their challenges and limitations are also summarized.

2.1 Classical Statistics-based NTP Models

In this chapter, we first introduce two representative statistics-based models, ARMA model and ARIMA model, which are widely used in time series forecasting problems.

2.1.1 ARMA Model

The ARMA(p, q) model is regarded as a linear combination of p-order autoregressive (AR) model and q-order moving average (MA) model, and is suitable for stationary time series data [17][18]. If $p = 0$, the ARMA model degrades into an MA model, and if $q = 0$, the

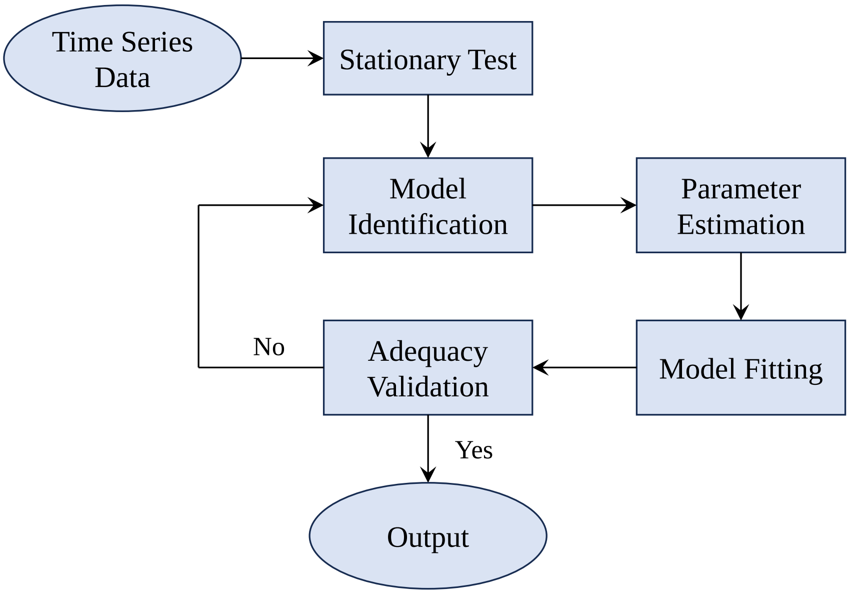


Fig. 2.1 Modelling process for the ARMA model.

ARMA model degenerates into an AR model [19]. Specifically, the AR part in the ARMA model depicts the correlation between current data values and past values, while the MA part represents the influence of the aggregated random errors from previous time points on the current observation. Hence, ARMA model is able to capture the statistical characteristics of the time series data accurately. The ARMA (p,q) model is formulated as:

$$\phi\left(B\right)Y_t = \theta\left(B\right)\varepsilon_t, \tag{2.1}$$

$$\phi\left(B\right) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p, \tag{2.2}$$

$$\theta\left(B\right) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q, \tag{2.3}$$

where Y_t and ε_t represent observation and white noise in moment t, respectively, while B is the back-shift operator [18] and is expressed as follows

$$\begin{aligned} B^i Y_t &= Y_{t-i}, \\ B^i \varepsilon_t &= \varepsilon_{t-i}. \end{aligned} \tag{2.4}$$

The modelling procedure of the ARMA model mainly includes 5 steps as shown in Fig. 2.1. The first step is stationary test, since the ARMA model requires stationary time series data as its input. Secondly, model identification is performed to obtain preliminary values of AR p and MA order q. On the basis of acquiring p and q, the ARMA model's parameters, $\phi_j, j \in [1,p]$ and $\theta_k, k \in [1,q]$, can be estimated by methods like maximum likelihood and least squares [20]. In the fourth and fifth steps, the ARMA model is fitted into the input data with the estimated parameters, and then it is necessary to validate that whether the residual sequence meets the white noise assumption. What is more, the goodness of fit can be tested according to Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC). The model with minimum AIC or BIC values is selected as the best model. When the model passes the adequacy validation, it can be used to perform prediction task, otherwise backing to model identification and repeating the modelling process to obtain an appropriate model.

2.1.2 ARIMA Model

The ARIMA model, referred to as the Box-Jenkins model, stands as a prominent paradigm within statistical models for time-series forecast [6]. Similar to the ARMA model, the ARIMA model also consists of the AR model and the MA model. The most significant difference is that compared with the ARMA model, the ARIMA model introduces a difference-stationary process, which enables the prediction of non-stationary time sequences [21][22]. Due to the existence of the difference-stationary process, the prediction results of the ARIMA model need to be restored through an inverse difference process. Fig. 2.2 illustrates the modelling process of the ARIMA model where the yellow boxes emphasize the difference between the ARIMA model and ARMA model. The following equations formulate the ARIMA model.

$$\phi(B)\nabla^d Y_t = \theta(B)\varepsilon_t, \tag{2.5}$$

where $\nabla^d = Y_t - Y_{t-d}$, represents the d-order difference, and the parameters $\phi, \theta, B, \varepsilon$ are the same as the parameters in ARMA model.

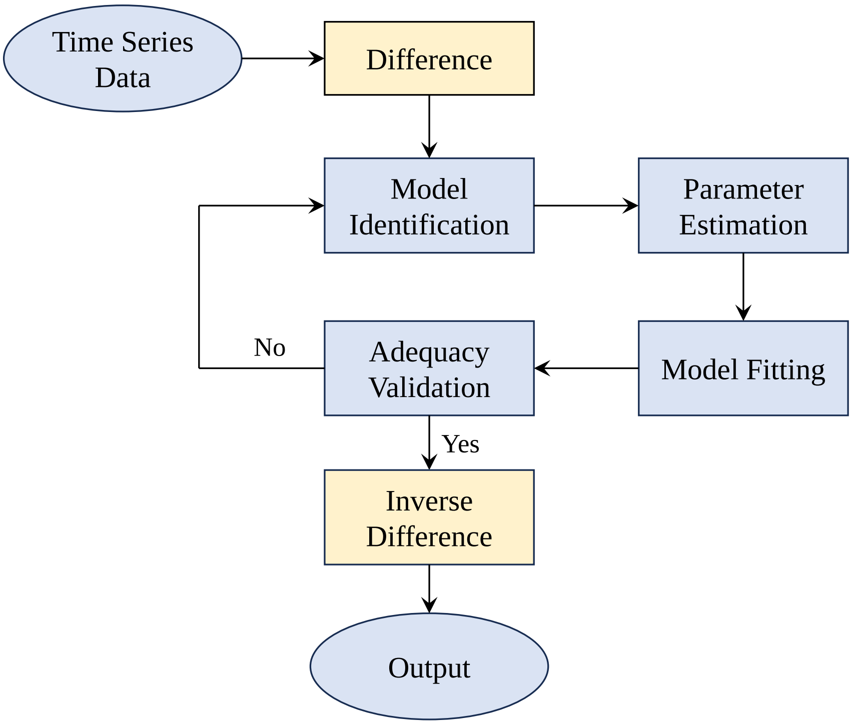


Fig. 2.2 Modelling process for ARIMA model.

In general, both the ARMA model and the ARIMA model aim to minimize the residual sequence by adjusting $\phi_j, j \in [1,p]$ and $\theta_k, k \in [1,q]$, thereby making the model fit the historical data well and accurate for prediction task.

2.2 Deep Learning-based NTP Models

Compared to traditional statistical-based methods, ML-based methods represented by deep neural networks show superior performance in terms of prediction accuracy [24]. Multiple activation functions enable the learning of complex nonlinear relationships. This section starts form the basic concepts like neuron, and then reviews two representative deep learning-based methods, the MLP network and the LSTM network.

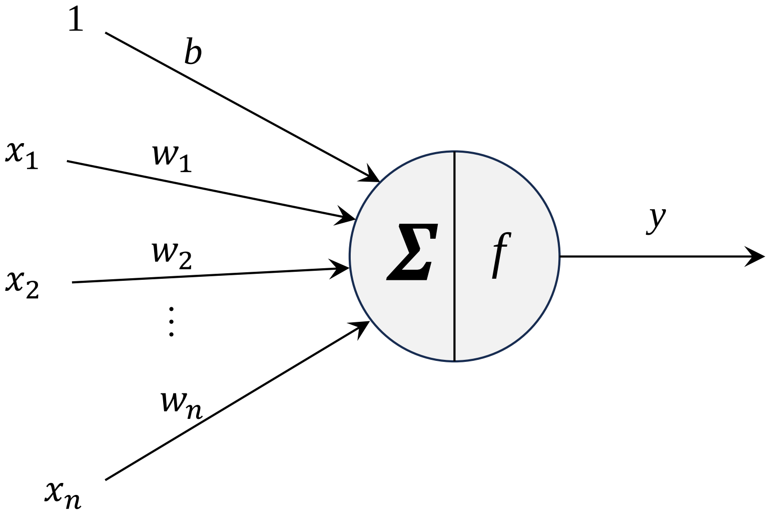


Fig. 2.3 The structure of the neuron.

2.2.1 Basic Concepts

The neuron is a fundamental concept and was first proposed by McCulloch and Pitts in 1943 [25], and can be expressed as

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right), \tag{2.6}$$

where x_i and w_i represent the i-th input and the corresponding weight, b and $f(\cdot)$ refer to the bias function and activation function, respectively, and y is the output of the neuron. As shown in Fig. 2.3, the neuron computes a weighted sum of input signals and bias, then passes the weighted sum through an activation function to get the final output. Each neuron has its own weights and bias, and these parameters are learned and adjusted gradually during training process, thereby improving the performance of the model. In addition, the activation function f is vital, which introduces nonlinear transformation, and thus enables neural networks to capture complex nonlinear features of historical data.

The commonly used activation functions include the sigmoid function, rectified linear unit (ReLU) function, and hyperbolic tangent (Tanh) function, which are represented as the following equations [26][27][28].

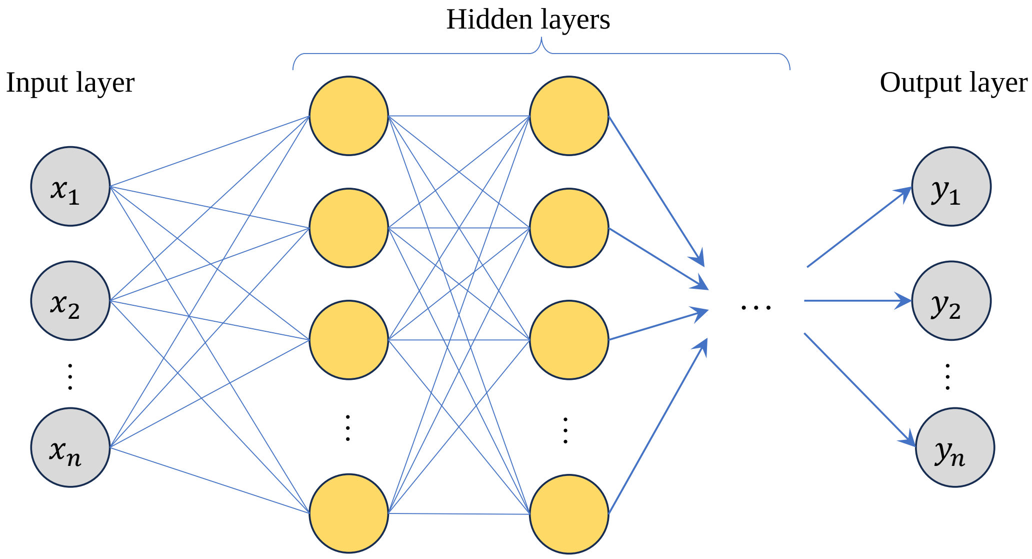


Fig. 2.4 The structure of the MLP network.

- Sigmoid function:

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}}. \tag{2.7}$$

- ReLU function [29]:

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{2.8}$$

- Tanh function:

$$f(x) = \frac{\sinh x}{\cosh x} = \frac{\mathrm{e}^{x}-\mathrm{e}^{-x}}{\mathrm{e}^{x}+\mathrm{e}^{-x}}. \tag{2.9}$$

2.2.2 MLP Network

The MLP network is a classical model, which possesses a relatively simple structure in comparison with other deep learning-based methods like LSTM network and Transformer network. As shown in Fig. 2.4 [30], the MLP network consists of an input layer, an output layer, and multiple hidden layers. Each hidden layer contains a number of neurons, and each neuron is connected to all of neurons in the neighboring layers. By means of this hierarchical structure, the neuron, as well as the activation function, the MLP network performs feature

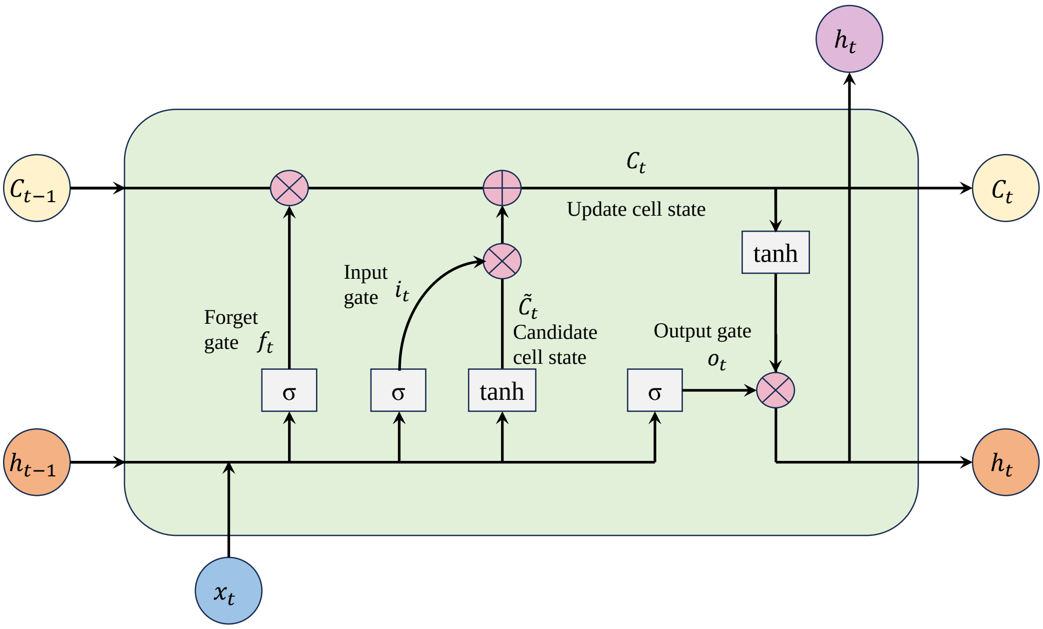


Fig. 2.5 The structure of the LSTM memory cell.

extraction on the input data layer-by-layer following forward propagation, then the error back propagation is utilized to minimize the loss function [31], thereby updating weight matrix and biases of the network.

2.2.3 LSTM Network

The LSTM network is an improvement of RNN, and was first proposed by Hochreiter and Schmidhuber in 1997 [32]. The LSTM memory cell structure is designed to help the network avoid the vanishing gradient problem [33]. As shown in Fig. 2.5, the LSTM cell contains three gate structures, i.e., a forget gate, an input gate, and an output gate [34][35][36].

The forget gate is the most important component in the LSTM cell [37], and determines what information needs to be discarded from cell state at the previous moment. The forget gate is formulated as

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{2.10}$$

where x_t and h_{t-1} are the inputs of the forget gate at moment t, denoting the input vector at time t and the hidden state at the previous time $t-1$, respectively. $W_{(\cdot)}$ and $b_{(\cdot)}$ represent the corresponding weight matrices and bias vectors, respectively. The forget gate adopts the sigmoid activation function $\sigma(\cdot)$ to output a vector of values within $[0,1]$, which indicates the degree of information retention of the previous cell state C_{t-1}. The input gate shares a similar structure with the forget, and can be expressed as

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right). \tag{2.11}$$

The input gate determines what information need to be added into the current cell state. Hence, the output of the input gate is acting on the candidate cell state $\tilde{C}_t$. It is defined as

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right). \tag{2.12}$$

With f_t, C_{t-1}, i_t, and $\tilde{C}_t$, the current cell state C_t is updated as

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{2.13}$$

where $\odot$ is the element-wise Hadamard product. The output gate is formulated as

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right). \tag{2.14}$$

Then the current hidden state can be updated.

$$h_t = o_t \odot \tanh\left(C_t\right). \tag{2.15}$$

By utilizing the memory cell, the LSTM network is able to learn long-term dependencies. Nowadays, the LSTM network and its variants have been widely applied in many research areas such as IoT [38] and natural language processing [39].

2.3 Related Works of NTP

NTP technology has a great potential to improve the network resource efficiency, while the extent to which its potential can be realized depends on its prediction performance [40]. Over the past few decades, the academia has made great effort in the NTP field, and the mainstream of NTP technology has undergone an evolution from the statistics-based method to the ML-based method [4] [5] [6] [41].

2.3.1 Statistics-based NTP

In traditional statistics-based models, historical network traffic data is fitted into some statistics or probability distributions to extract traffic patterns and attributes, such as the α-stable model, the ARMA model, the ARIMA model, the Seasonal Autoregressive Integrated Moving Average (SARIMA) model, etc.

Authors in [42] focused on the self-similar network traffic and utilized the the theory of α-stable processes to extract the traffic patterns in mobile network. What is more, as the representation for traditional statistics-based models, the ARMA and ARIMA model are often used to extract linear features from historical traffic data [6]. As mentioned above, the ARMA model is suitable for stationary sequences. L. Tang *et al.* employed the ARMA model to predict the future load state of virtual networks, and then proposed a dynamic resource allocation scheme for virtual networks based on the prediction results [43]. While the ARIMA model fits non-stationary sequences well by adding a difference-stationary process [21]. In [22], the ARIMA model was used to predict the normal traffic in the next minute to identify DoS and DDoS attacks.

In addition, the variants of the ARIMA model are also widely used in NTP field. For example, the SARIMA model considered the seasonal correlations in the variations of network traffic stream [44]. F. Xu *et al.* in [45] adopted the SARIMA to forecast the seasonal components in cellular traffic data. Besides, some nonlinear models were proposed for nonlinear features, such as the GARCH model [46]. To further improve the performance, an ARIMA prediction model aided with entropy theory was proposed in [47]. Besides the above models, the ON-OFF model [48], the Kalman function [49], the covariance function [50], and the Holt-Winter's exponential smoothing model [51] were also introduced to fit the temporal-spatial characteristics of mobile traffic data.

Although traditional statistics-based models have the advantage of low computational complexity and do not involve the complicated hyper-parameter optimization problem, in general, they do not perform very well in prediction accuracy in comparison to ML-based models. In [24], the authors compared the ARIMA model with the LSTM network, and concluded that the LSTM network outperforms the ARIMA model. Similar conclusion was made in [23] that the artificial neural network is superior to the linear models like the ARMA model and the Holt-Winters algorithm.

2.3.2 Shallow Learning-based NTP

Shallow learning is a concept opposite to deep learning [52], which mainly refers to traditional ML algorithms such as gaussian process (GP), linear regression (LR), and support vector machine (SVM) [53]. Numerous shallow learning algorithms, like support vector regression (SVR) [54] [55] [56], LR [57] [58], GP [59] [60] [61], and principal component analyses [62] were proposed to conduct NTP.

SVR, a variant of SVM, is an application of SVM to the regression task [63]. Authors in [54] proposed a platform for collecting and predicting the real-time cellular traffic data, where the SVR algorithm was used for real-time cellular traffic prediction. Moreover, the authors utilized the grid search to optimize the hyper-parameters of the SVR model in consideration of the fact that hyper-parameters have a huge impact on the generalization performance of the SVR model. In [55], the SVR model was used for the prediction of telephone traffic, and an improved grid search method was used for hyper-parameter selection. In [56], Y. Zhang *et al.* proposed a SVR model to predict the voice traffic data of the base stations in global system for mobile communications networks. Historical traffic data generated by the target base station and neighbouring base stations were used as inputs to the SVR model, which improved the prediction accuracy. Finally, the particle swarm optimization (PSO) algorithm was adopted to optimize the SVR model's hyper-parameters.

In [57], a hybrid LR-based traffic prediction model was proposed for the real-world traffic data provided by Deutsche Telekom AG, Germany,and the effect of window size on model performance was investigated. Based on the prediction results, a power management system was proposed to automatically turn on/off base stations to minimize energy consumption while ensuring quality of service (QoS). Authors in [58] proposed a joint NTP model combining LR and Random Forest (RF) for backbone optical networks. The experimental results proved that the joint NTP model has higher prediction accuracy than the standalone models.

Authors in [59] proposed a GP-based NTP model for the cloud radio access network and utilized the alternating direction method of multipliers to optimize the model's hyper-parameters. In [60], a GP-based model with the grid spectral mixture (GSM) kernel function was used for 5G traffic prediction, and an adaptive GSM kernel learning algorithm was proposed to obtain the corresponding kernel configuration. Authors in [61] adopted three shallow learning-based models, SVM, Gaussian Process Regression and Robust Linear Regression to predict the key parameters in LTE networks, and validated the feasibility of these NTP models through experiments.

In general,these shallow learning-based methods outperform the statistics-based methods in prediction accuracy, since they are better able to capture linear and nonlinear features in network traffic data and have stronger learning capabilities. However, these methods are not as interpretable as statistics-based methods because their parameters cannot be used directly to interpret the prediction results. In addition, these shallow learning-based methods have already involved hyper-parameter selection problem.

2.3.3 Deep Learning-based NTP

In recent years, the deep learning technology has been leveraged in wireless NTP. In comparison to the shallow learning-based methods, the deep learning-based methods include more layers (typically more than 3 layers) and introduce a more complex structure like gate structure and self-attention mechanism, thereby exhibiting significantly enhanced learning capacity.

L. Nie *et al.* in [64] proposed a deep belief network and Gaussian models (DBNG) to predict the cell-level wireless network traffic loads. In [66], a multi-scale deep echo-state network (ESN) based prediction model was proposed to learn the trends and characteristics of network traffic at different temporal scales. Z. Wang *et al.* [65] integrated the network spatial information with the cell-level wireless network traffic load series and proposed a graph neural network (GNN) based NTP model.

Authors in [67] conducted both single-step and multi-step prediction for mobile traffic in LTE base stations, and conclude that the prediction error increases versus prediction step size. Moreover, in comparison with the ARIMA model and feed-forward neural network (FFNN), LSTM network proved to be the most effective model. Similar conclusion was made in [68], and the authors compared the LSTM network, FFNN, and ARIMA model, then validated that the LSTM network possess higher prediction accuracy and faster convergence speed. Z. Wang *et al.* in [69] utilized generative adversarial networks to generate traffic data for privacy protection while increasing prediction accuracy, and the LSTM network was adopted to realize the multi-step prediction.

Authors in [70] considered the temporal and spatial correlations between neighbouring base stations, and proposed a hybrid deep learning model for LTE networks of China Mobile at Suzhou, where the LSTM networks and autoencoder-based deep models are used for temporal and spatial modeling, respectively. Experimental results indicated that the hybrid deep learning model is the most effective compared to the SVR and ARIMA models. In [71], C. Qiu *et al.* proposed a multi-task learning approach based on the LSTM network to improve the model's prediction accuracy by exploring the similarities and differences of traffic patterns among neighbouring mobile cells. With reduced connection probability between neurons, a random connectivity LSTM network based traffic prediction model was proposed in [72] to decrease the model's training complexity.

Regarding historical traffic loads generated in multiple base stations as the inputs, the CNN and convolutional LSTM network based NTP models were proposed in [73] and [74], respectively. C. Zhang *et al.* in [75] proposed a densely connected convolutional neural network (CNN) combined with a parameter matrix-based scheme to predict the traffic of short message service (SMS) and Call service in Milan, 2013. In [76], a spatial-temporal cross-domain neural network model based on convolutional LSTM is proposed for predicting SMS, call and internet service data in Milan, 2013. Furthermore, based on transfer learning, a fusion transfer strategy is proposed to share parameters among different models to improve the prediction accuracy. Authors in [77] proposed to decompose the network traffic load series into several product function components using the local mean decomposition method, each of which is then predicted with a bidirectional LSTM network model.

Y. Hu *et al.* [78] proposed to utilize attention mechanism to depict the spatial-temporal characteristics of wireless traffic patterns and presented a transformer based prediction model. By combining attention and convolution mechanisms into traffic analysis, a multi-view spatial-temporal graph network based prediction model was proposed in [79] to learn diverse global spatial-temporal dependencies of cellular traffic loads.

These research works exploited the DL technology to mine the hidden characteristics of wireless network traffic patterns and achieved the state-of-the-art accuracy performance. However, they mainly focused on designing elaborate prediction models/algorithms for different NTP tasks and simply mentioned the hyper-parameter settings they used without providing an explanation of the reasons for the settings. Moreover, they ignored how to find the optimal hyper-parameters for each predict model based on the corresponding prediction task's intrinsic characteristics or hyper-parameter selection experience accumulated from other prediction tasks. In [80], the authors made some initial attempts to elevate a new cell-level traffic prediction model's performance by providing it the proper initial weight vector on the basis of initial weight vector selection strategies of previous prediction models. Nevertheless, optimizing the prediction models' hyper-parameters has also not been addressed.

2.3.4 Conclusion

Overall, traditional statistics-based models are not specifically designed for NTP tasks, and their limited learning capacity often results in inferior predictive accuracy. Shallow learning-based models, while capable of capturing some nonlinear relationships, are similarly constrained by their limited learning capacity, making it challenging to achieve the most accurate predictions. In contrast, deep learning-based models excel in feature extraction and generally deliver high predictive accuracy. However, they involve high computational costs, and lack interpretability.

Chapter 3

Analytic Network Traffic Prediction Based on User Behavior Modeling

Overview

In this chapter, an interpretable UBB NTP method is proposed. Based on user behavior, a weekly traffic demand profile can be naturally sorted into three categories, i.e., weekday, Saturday, and Sunday. For each category, the traffic pattern is divided into three components which are mainly generated in three time periods, i.e., morning, afternoon, and evening. Each component is modeled as a normal-distributed signal. Numerical results indicate the UBB NTP method matches the practical wireless traffic demand very well. Compared with existing methods, the proposed UBB NTP method improves the computational efficiency and increases the predictive accuracy.

3.1 Introduction

AS mentioned in chapter 1, the explosive growth in users' network traffic demand leads to a severe challenge in network efficiency. As one of the most promising NTMA technologies to improve network resource allocation, NTP has garnered widespread attention within the academic community.

It has been found that traffic patterns are closely related to land use types, such as commercial and residential areas [81][82][83]. However, the state-of-the-art relevant to NTP only focuses on historical traffic data and does not take into account the connection between traffic data and the real world regardless of the statistics-based methods and the ML-based methods. This strategy affects the performance of statistics-based methods in terms of prediction accuracy and significantly constrains the interpretability and computational efficiency of ML-based method.

It is vital to note that traffic patterns are present in historical traffic data and, more importantly, that user behavior dominates the patterns. The state-of-the-art is built on the former and overlooks the latter. They extract an aggregation of traffic patterns from historical data and store it with a specific model, which is comprehensible for computer, but meaningless and invisible for human being. Generally, they lack explanatory power regardless of how accurate and complicated they are. This work jumps out of the shackle of existing works. To enhance the overall performance, user behavior characteristics is employed in the UBB NTP framework. Overall traffic profile is regarded as the superposition of several normal-distributed signals and the specific parameters are extracted from real-world traffic data. Numerical results show that it is a simple, efficient, accurate, and highly interpretable method to discover traffic patterns.

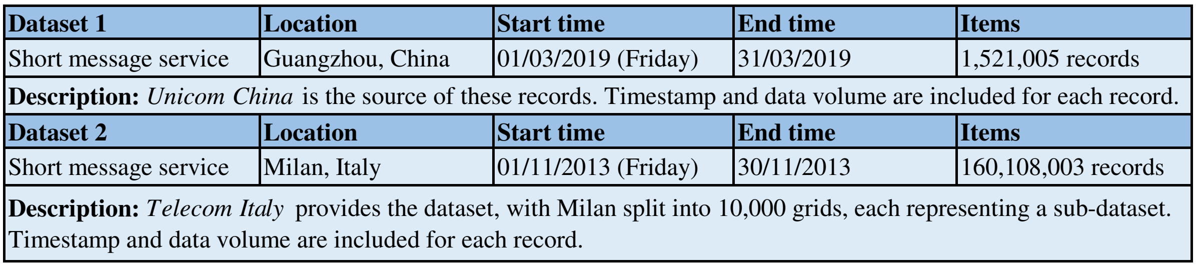
In general, the proposed method effectively utilizes the principal status of user behavior, and achieves the advantages as follows:

1) Compared with existing methods, our method provides an interpretable NTP solution which is visual and comprehensible.

2) Our method has a significant advantage in terms of computational efficiency.

3) Our method establishes a correspondence between model parameters and user habits. The compatible expression of model parameters provides an opportunity to compare traffic patterns in different regions.

The rest of the chapter is structured as follows. Section 3.2 details the key points of the UBB NTP method and establish the mathematical model. Section 3.3 simulates the UBB



<table>
<tr><th>Dataset 1</th><th>Location</th><th>Start time</th><th>End time</th><th>Items</th></tr>
<tr><td>Short message service</td><td>Guangzhou, China</td><td>01/03/2019 (Friday)</td><td>31/03/2019</td><td>1,521,005 records</td></tr>
<tr><td colspan="5"><b>Description:</b> <i>Unicom China</i> is the source of these records. Timestamp and data volume are included for each record.</td></tr>
<tr><td><b>Dataset 2</b></td><td><b>Location</b></td><td><b>Start time</b></td><td><b>End time</b></td><td><b>Items</b></td></tr>
<tr><td>Short message service</td><td>Milan, Italy</td><td>01/11/2013 (Friday)</td><td>30/11/2013</td><td>160,108,003 records</td></tr>
<tr><td colspan="5"><b>Description:</b> <i>Telecom Italy</i> provides the dataset, with Milan split into 10,000 grids, each representing a sub-dataset. Timestamp and data volume are included for each record.</td></tr>
</table>

Fig. 3.1 Description of the two adopted datasets.

NTP method and benchmark methods, and performs a comparative analysis. Finally, the conclusions and future plans are drawn in Section 3.4.

3.2 The Proposed UBB NTP Method

There are two approaches to gather user habits: one is from daily behavior, and the other is through real-world traffic data. Based on these habits, we build the mathematical model.

3.2.1 Analysis for Daily Behavior

Generally, people are accustomed to being active during the day and sleeping at night. In addition to sleeping hours, every day includes three main periods, i.e., morning, afternoon, and evening. Most people carry out their daily activities throughout these three periods. For example, every morning and afternoon on weekdays are generally working hours, whereas every evening is typically for recreation. These three periods thus correlate to the busiest times for network services and the traffic pattern fits well with the daily routine of users [84]. Hence, we divide the daily traffic into three traffic components, corresponding to the three time periods.

Furthermore, in the absence of prior information, we conceive that the peak time of each traffic component represents the time preference of users to use network. Due to various subjective or objective factors, some users' traffic demands deviate from the time preference. The greater the magnitude of traffic deviation, the lower the probability of occurrence. This assumption is well-founded because procrastination is prevalent in the

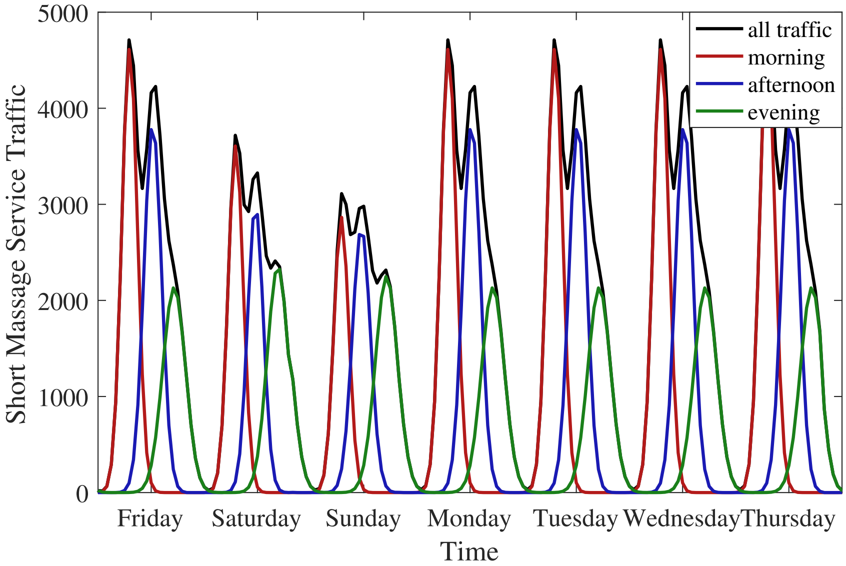


Fig. 3.2 Traffic distribution and the components in the morning, afternoon and evening.

population [85][86][87]. Statistically, the prevalence of procrastination is as high as 20-25% in the general population [86] and 15% of adults suffer from severe procrastination [87]. Correspondingly, there will be some users who like to finish their tasks in advance. In addition, outside the three main periods, the traffic caused by users' whims takes up a tiny percentage. For simplicity, all of the traffic outside the three main periods is regarded as an extension of these traffic components.

3.2.2 Analysis for Traffic Data

The datasets are collected from Guangzhou, China, and Milan, Italy. The specifics of these datasets are shown in Fig. 3.1. Meanwhile, Dataset 1 is purchased and not publicly accessible, while Dataset 2 is available [88]. Although SMS data may constitute only a small portion of current network traffic, studying SMS traffic remains of significant importance. Firstly, SMS services are widely used globally, particularly in emergency communications. Even in today's world of surging data traffic, SMS remains a crucial component of many essential communication services. Secondly, SMS traffic significantly impacts the stability and reliability of network infrastructure. By researching SMS traffic, we can ensure that networks operate reliably during peak periods or emergencies. Thirdly, SMS services have a wealth of historical data. This data can be utilized to analyze communication patterns, predict traffic trends, and provide valuable insights for network optimization. Finally, studying SMS traffic lays the groundwork for more complex traffic management in the future. Understanding and optimizing simpler types of traffic is instrumental in developing more efficient network management techniques and strategies.

In both Datasets 1 and 2, the network traffic has a periodic variation during a week, which is consistent with the common sense that the week is a natural cycle of human activity. In addition, the traffic data on weekdays show a similar trend. It is mainly because the bulk of the urban populations leads a highly repetitive life on weekdays. The representative groups include students, teachers, enterprise employees, government officers, and so on. Hence, the workdays are characterized by the same traffic components. Moreover, the employees in several occupations, like express industry, food service, etc., work seven days a week. Therefore, a portion of people maintains the pace of life on weekends. However, the traffic trends on Saturday and Sunday are different. Saturday is the end of the workweek for most users, prompting participation in social activities, shopping, entertainment, or personal hobbies. As the first day of the weekend, activities on Saturday often extend into the evening, with a greater inclination towards outings and gatherings. In contrast, although Sunday is also a non-working day, its proximity to the upcoming workweek encourages rest, household chores, or preparation for the week ahead. The social and business environment also plays a role. For instance, many stores have extended hours on Saturdays to accommodate shoppers, while shorter Sunday hours contribute to variations in activities between the two days. Consequently, Saturday and Sunday tend to exhibit different traffic patterns. Therefore, we build dedicated models for Saturday and Sunday, respectively.

As shown in Fig. 3.2, we adopt three categories to represent the traffic on weekdays, Saturdays, and Sundays, respectively. Combined with the three main periods mentioned above, a total of nine traffic components are required to construct the UBB NTP method. The abbreviations in Table 3.1 represent these components. The overall traffic is the superposition of all traffic components. It is crucial to note that daily traffic could be represented by more

Table 3.1 The symbols corresponding to the 9 traffic components.

	Weekday	Saturday	Sunday
Morning	mw	msa	msu
Afternoon	aw	asa	asu
Evening	ew	esa	esu

or fewer components if there are sufficient social science data to back it up. It is a reflection of the scalability of the UBB NTP method.

3.2.3 Mathematical Model

We adopt the SMS data in Guangzhou and Milan to extract parameters for mathematical model, respectively. As shown in Fig. 3.2, the daily traffic curve is converted into three components. The red curve represents the traffic component distributed on the entire time axis with morning traffic as the main body. Similarly, the blue and green ones correspond to the afternoon and evening components, respectively.

As some users prefer to finish their work a little early, while others tend to procrastinate, it is assumed that the time preferences of different users follow an independent identically distribution (i.i.d.). According to the central-limited theorem, the average time preference follows a normal distribution. Therefore, the distribution of users with different time preferences can be approximately characterized by a Gaussian signal. The morning traffic component on a certain workday is expressed as:

$$G_{\mathrm{mw}}(t) = R_{\mathrm{mw,p}}\exp\left(-\frac{\left(t - t_{\mathrm{mw,p}}\right)^2}{2\sigma_{\mathrm{mw}}^2}\right), \tag{3.1}$$

where $t_{\mathrm{mw,p}}$ is the time when each message is expected to be sent, σ_{mw}^2 is the variance and $R_{\mathrm{mw,p}}$ is the peak value. Similarly, each traffic component can be represented by:

$$G_{\mathrm{c}}(t) = R_{\mathrm{c,p}}\exp\left(-\frac{\left(t - t_{\mathrm{c,p}}\right)^2}{2\sigma_{\mathrm{c}}^2}\right), \tag{3.2}$$

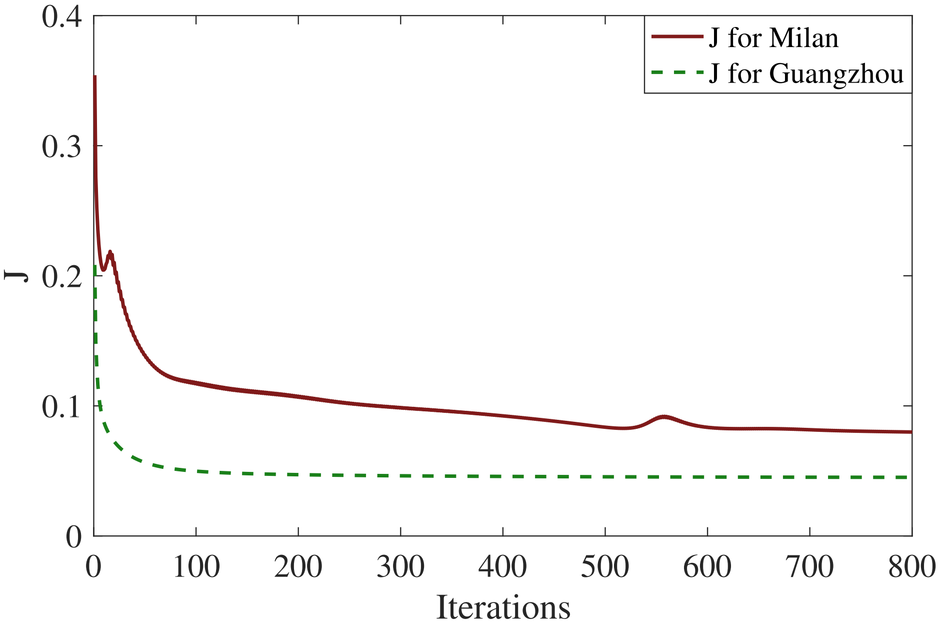


Fig. 3.3 J varies with the number of iterations.

in which t is in a 24-hour format, $\mathrm{c} \in \{\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3\}$, represent weekday, Saturday and Sunday, respectively, and $\mathrm{c}_1 \in \{\mathrm{mw}, \mathrm{aw}, \mathrm{ew}\}$, $\mathrm{c}_2 \in \{\mathrm{msa}, \mathrm{asa}, \mathrm{esa}\}$, $\mathrm{c}_3 \in \{\mathrm{msu}, \mathrm{asu}, \mathrm{esu}\}$. These abbreviations are shown in Table 3.1. Therefore, the hourly traffic at time t, the kth day of a week, can be represented as:

$$Y_k(t) = \sum_{n_\mathrm{w}} \begin{bmatrix} \sum\limits_{n_\mathrm{d}=1}^{5} \sum\limits_{\mathrm{c}_1} G_{\mathrm{c}_1}\left(t + 24\left(k - n_\mathrm{d}\right) + 168 n_\mathrm{w}\right) \\ +\sum\limits_{\mathrm{c}_2} G_{\mathrm{c}_2}\left(t + 24(k-6) + 168 n_\mathrm{w}\right) \\ +\sum\limits_{\mathrm{c}_3} G_{\mathrm{c}_3}\left(t + 24(k-7) + 168 n_\mathrm{w}\right) \end{bmatrix}, \tag{3.3}$$

with the index of the day $k \in \{1,2,3,4,5,6,7\}$, the index of the week number $n_\mathrm{w} \in (-\infty, +\infty)$, the index of the weekday n_d. After exchanging the sum order, we have:

$$\begin{aligned} Y_k(t) = &\sum_{\mathrm{c}_1} \sum_{n_\mathrm{d}=1}^{5} \sum_{n_\mathrm{w}} G_{\mathrm{c}_1}\left(t + 24\left(k - n_\mathrm{d}\right) + 168 n_\mathrm{w}\right) \\ &+\sum_{\mathrm{c}_2} \sum_{n_\mathrm{w}} G_{\mathrm{c}_2}\left(t + 24\left(k - 6\right) + 168 n_\mathrm{w}\right) \\ &+\sum_{\mathrm{c}_3} \sum_{n_\mathrm{w}} G_{\mathrm{c}_3}\left(t + 24\left(k - 7\right) + 168 n_\mathrm{w}\right), \end{aligned} \tag{3.4}$$

$$\begin{aligned} Y_k(t) = &\sum_{\mathrm{c}_1} R_{\mathrm{c}_1,\mathrm{p}} \sum_{n_\mathrm{d}=1}^{5} \sum_{n_\mathrm{w}} \exp\left(-\frac{\left(t+24(k-n_\mathrm{d})-t_{\mathrm{c}_1,\mathrm{p}}+168n_\mathrm{w}\right)^2}{2\sigma_{\mathrm{c}_1}^2}\right) \\ &+\sum_{\mathrm{c}_2} R_{\mathrm{c}_2,\mathrm{p}} \sum_{n_\mathrm{w}} \exp\left(-\frac{\left(t+24(k-6)-t_{\mathrm{c}_2,\mathrm{p}}+168n_\mathrm{w}\right)^2}{2\sigma_{\mathrm{c}_2}^2}\right) \\ &+\sum_{\mathrm{c}_3} R_{\mathrm{c}_3,\mathrm{p}} \sum_{n_\mathrm{w}} \exp\left(-\frac{\left(t+24(k-7)-t_{\mathrm{c}_3,\mathrm{p}}+168n_\mathrm{w}\right)^2}{2\sigma_{\mathrm{c}_3}^2}\right), \end{aligned} \tag{3.5}$$

where n_w is the domain of Gaussian signal. For simplicity, we restrict the range of n_w to [-1, 1]. Then, we have:

$$\begin{aligned} Y_k(t) \approx &\sum_{\mathrm{c}_1} R_{\mathrm{c}_1,\mathrm{p}} \sum_{n_\mathrm{d}=1}^{5} \sum_{n_\mathrm{w}=-1}^{+1} \exp\left(-\frac{\left(t+24(k-n_\mathrm{d})-t_{\mathrm{c}_1,\mathrm{p}}+168n_\mathrm{w}\right)^2}{2\sigma_{\mathrm{c}_1}^2}\right) \\ &+\sum_{\mathrm{c}_2} R_{\mathrm{c}_2,\mathrm{p}} \sum_{n_\mathrm{w}=-1}^{+1} \exp\left(-\frac{\left(t+24(k-6)-t_{\mathrm{c}_2,\mathrm{p}}+168n_\mathrm{w}\right)^2}{2\sigma_{\mathrm{c}_2}^2}\right) \\ &+\sum_{\mathrm{c}_3} R_{\mathrm{c}_3,\mathrm{p}} \sum_{n_\mathrm{w}=-1}^{+1} \exp\left(-\frac{\left(t+24(k-7)-t_{\mathrm{c}_3,\mathrm{p}}+168n_\mathrm{w}\right)^2}{2\sigma_{\mathrm{c}_3}^2}\right). \end{aligned} \tag{3.6}$$

Thus, the parameter estimation problem becomes an optimization problem:

$$\underset{\substack{R_{\mathrm{c,p}},\, t_{\mathrm{c,p}},\, \sigma_\mathrm{c}^2 \\ \mathrm{c} \in \left\{\substack{\mathrm{mw,aw,ew,} \\ \mathrm{msa,asa,esa,} \\ \mathrm{msu,asu,esu}}\right\}}}{\text{minimise}} \quad J, \tag{3.7}$$

where $J = ||Y_k(t) - Y_\mathrm{meas}||^2$ and Y_meas refers to the vector consisting of traffic measurements. There are many ways to solve this problem. This work adopts a simple gradient descent method. As shown in Fig. 3.3, J gradually decreases and converges as the number of iterations increases.

3.2.4 Initial Parameter Selection Strategy

The initial parameter set of the proposed UBB NTP method is vital to its performance, and contains the parameters of nine normal-distributed signals related to the nine traffic components listed in Table 3.1.

As discussed in section 3.2, the daily traffic is regarded as a superposition of three traffic component corresponding to the three time periods of a day. Then traffic related to weekday G_weekday can be represented by

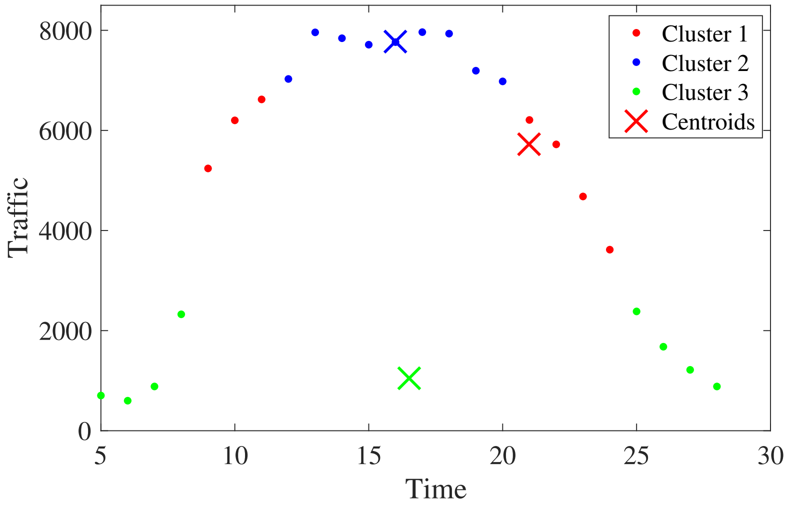


Fig. 3.4 The clusters and centroids generated by the K-means clustering algorithm.

$$G_{\text{weekday}}(t) = G_{\text{mw}}(t) + G_{\text{aw}}(t) + G_{\text{ew}}(t), \tag{3.8}$$

where $G_{\text{mw}}(t)$, $G_{\text{aw}}(t)$, and $G_{\text{ew}}(t)$ refer to the three traffic components of the weekday corresponding to morning, afternoon, and evening, respectively. $G_{\text{weekday}}(t)$ contains 24 data points as the time granularity is one hour. Each data point $p = [a,\ b]$, where a denotes the time t and $b = G_{\text{weekday}}(t)$ denotes the traffic value at time t. Different data points have varying degrees of impact on the three traffic components, and this section aims to calculate the initial parameters of each traffic component.

In this section, we propose two approaches to solve the initial parameter selection problem. The first one is a K-means clustering based approach. The second approach is the one we designed specifically for the initial parameter selection problem of the UBB NTP model, referred to as analytical calculations.

The K-means clustering is a common unsupervised learning algorithm to automatically divide the data points into different clusters. It minimizes the variance of the data points within clusters. In this process, each data point is assigned to the cluster where the nearest cluster' centroid is located. Thus, the automatic clustering of data points is completed. Fig. 3.4 shows the result of the K-means clustering algorithm targeting R_{weekday}. Different colors

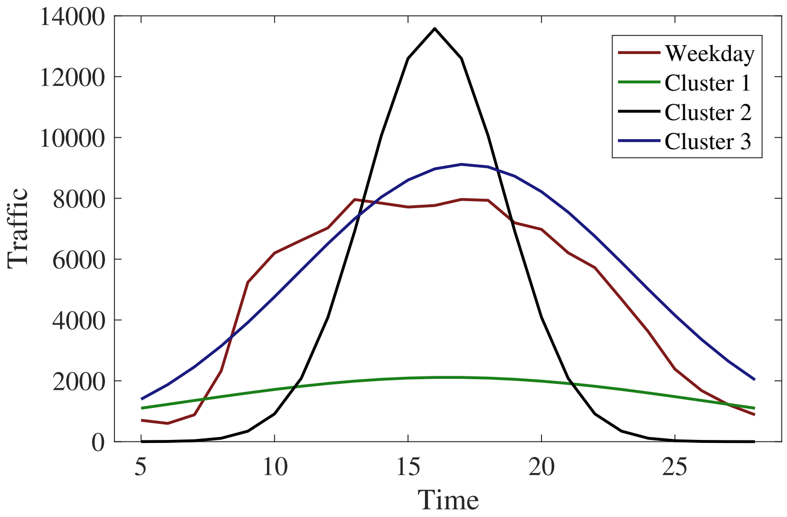


Fig. 3.5 The three traffic components corresponding to Cluster 1,2, and 3.

represent different clusters, and the "X" marks denote the centroids of these clusters. Cluster v can be represented by $C_v = \{p_1, p_2, \ldots, p_s\}$, where the character s is the cardinality of the subset C_v and $p_i = [a_i,\ b_i]$, $\forall i \in [1, s]$. Then the initial parameters of the traffic component regarding C_v can be calculated as

$$t_v = \frac{1}{s}\sum_{i=1}^{s}(a_i)\,, \tag{3.9}$$

$$\sigma_v^2 = \frac{1}{s}\sum_{i=1}^{s}(a_i - t_v)^2\,, \tag{3.10}$$

$$R_v = \frac{1}{s}\sum_{i=1}^{s} b_i \mathrm{exp}\left(\frac{(a_i - t_v)^2}{2\sigma_v^2}\right). \tag{3.11}$$

Then the initial parameters of Cluster 1, 2, and 3 are obtained with the above approach. Fig. 3.5 shows the traffic components corresponding to Cluster 1, 2, and 3. Fig. 3.6 plots the superposition of these traffic components. As shown in Fig. 3.6, there is a huge gap between the superposition and the real-world traffic data, which means the K-means clustering based approach is not well aligned with the problem of initial parameter selection.

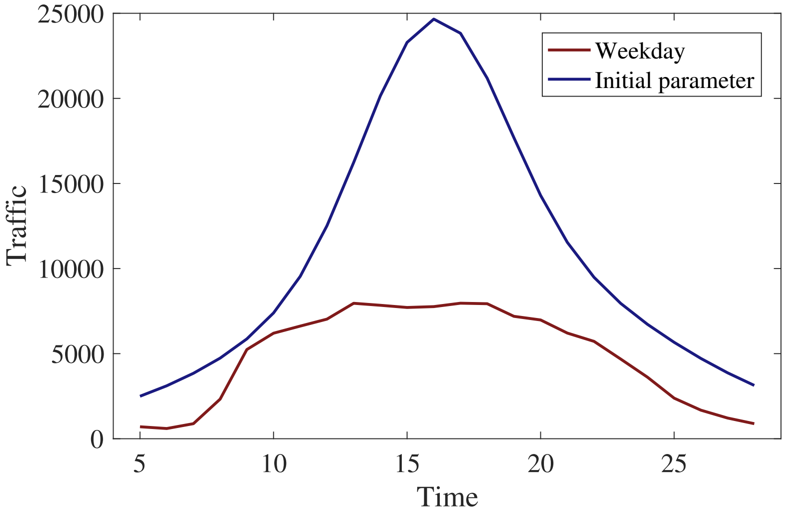


Fig. 3.6 The comparison between the real-world traffic data on weekday and the traffic profile associated with the initial parameters derived from the K-means clustering based approach.

Hence, we proposed the analytical calculations in the context. Since there is a clear chronological sequence between the daily traffic components in the UBB NTP model. Then we have $t_{mw,p} < t_{aw,p} < t_{ew,p}$. Furthermore, for the network traffic prior to moment $t_{mw,p}$, the earlier the traffic arises, the greater the contribution to $G_{\mathrm{mw}}(t)$. After the moment $t_{ew,p}$, the later the traffic arises, the greater the contribution to $G_{\mathrm{ew}}(t)$. The network traffic between the moments $t_{mw,p}$ and $t_{ew,p}$ is likely to be affected by the $G_{\mathrm{mw}}(t)$, $G_{\mathrm{aw}}(t)$, and $G_{\mathrm{ew}}(t)$, together. Therefore, we constructed two subsets, labeled as $\mathrm{Subset_m}$ and $\mathrm{Subset_e}$, containing data points from two time periods, morning and evening, respectively, based on the general definition of morning and evening. $\mathrm{Subset_m} = \{p_{m1}, p_{m2}, \dots, p_{m\alpha}\}$ consists of α data points occurring before 13:00 and $p_{mi} = [a_{mi}, b_{mi}]$, $\forall i \in [1, \alpha]$, while $\mathrm{Subset_e}$ contains β points after 18:00. Since $G_{\mathrm{mw}}(t)$ is primarily influenced by the data points in $\mathrm{Subset_m}$, we have the following approximation relation:

$$b_{mi} \approx R_m \exp\left(-\frac{(a_{mi} - t_m)^2}{2\sigma_m^2}\right). \tag{3.12}$$

By applying the natural logarithm to both sides of Formula 3.12, we obtain

$$\ln(b_{mi}) \approx \ln(R_m) - \frac{(a_{mi} - t_m)^2}{2\sigma_m^2}. \tag{3.13}$$

Three data points are chosen from Subset$_\mathrm{m}$, resulting in $\binom{\alpha}{3}$ possible selections. Iterating over the $\binom{\alpha}{3}$ possible selections and using (a_{m1}, b_{m1}), (a_{m2}, b_{m2}), (a_{m3}, b_{m3}) to denote the three randomly selected data points, we have

$$\ln(b_{m1}) \approx \ln(R_m) - \frac{(a_{m1} - t_m)^2}{2\sigma_m^2}, \tag{3.14}$$

$$\ln(b_{m2}) \approx \ln(R_m) - \frac{(a_{m2} - t_m)^2}{2\sigma_m^2}, \tag{3.15}$$

$$\ln(b_{m3}) \approx \ln(R_m) - \frac{(a_{m3} - t_m)^2}{2\sigma_m^2}. \tag{3.16}$$

Subtracting Formula 3.14 from Formula 3.15 yields

$$\ln\left(\frac{b_{m2}}{b_{m1}}\right) \approx \frac{1}{2\sigma_m^2}(a_{m1} - a_{m2})(a_{m1} + a_{m2} - 2t_m). \tag{3.17}$$

Subtracting Formula 3.15 from Formula 3.16 yields

$$\ln\left(\frac{b_{m3}}{b_{m2}}\right) \approx \frac{1}{2\sigma_m^2}(a_{m2} - a_{m3})(a_{m2} + a_{m3} - 2t_m). \tag{3.18}$$

Dividing Formula 3.18 by Formula 3.17 results in

$$\frac{[\ln(b_{m3}) - \ln(b_{m2})](a_{m1} - a_{m2})}{[\ln(b_{m2}) - \ln(b_{m1})](a_{m2} - a_{m3})} \approx \frac{a_{m2} + a_{m3} - 2t_m}{a_{m1} + a_{m2} - 2t_m}. \tag{3.19}$$

Let constant C represent the left-hand side of Formula 3.19. The initial parameter t_m can be calculated as

$$t_m \approx \frac{Ca_{m1} + (C-1)a_{m2} - a_{m3}}{2C - 2}. \tag{3.20}$$

Substituting t_m into Formula 3.17, the initial parameter σ_m^2 is given by

$$\sigma_m^2 \approx \frac{(a_{m1} - a_{m2})(a_{m1} + a_{m2} - 2t_m)}{2\ln(b_{m2}) - 2\ln(b_{m1})}. \tag{3.21}$$

Substituting t_m and σ_m^2 into Formula 3.14, the initial parameter R_m is given by

$$R_m = b_{m1}\exp\left(\frac{(a_{m1} - t_m)^2}{2\sigma_m^2}\right). \tag{3.22}$$

Thus, we can obtain a total of $\binom{\alpha}{3}$ initial parameter sets for $G_{mw}(t)$. Furthermore, the practical significance of t_m requires that $t_m > 0$. In addition, according to the 3-sigma rule of normal distribution, we have $6\sigma_m \leq 24$, which simplifies to $\sigma_m \leq 4$. Then we screen these initial parameter sets according to these constraints and obtain the candidate initial parameter set $\mathscr{P}_{mw}$ for $G_{mw}(t)$. With the same method, the candidate initial parameter set $\mathscr{P}_{ew}$ associated with $G_{ew}(t)$ can be required.

By iterating over all combinations of elements from sets $\mathscr{P}_{mw}$ and $\mathscr{P}_{ew}$, all the possible initial traffic components in morning and evening can be obtained, refer to as $\hat{G}_{mw}(t)$ and $\hat{G}_{ew}(t)$. Then the remaining traffic can be represented as

$$G_{\text{remain}}(t) = G_{\text{weekday}}(t) - \hat{G}_{mw}(t) - \hat{G}_{ew}(t), \tag{3.23}$$

which is used to derive the initial parameter sets $\mathscr{P}_{aw}$ for $G_{aw}(t)$ with the same method. What is more, the estimation error of the initial parameters can be expressed as

$$E_{\text{error}} = G_{\text{remain}}(t) - \hat{G}_{aw}(t), \tag{3.24}$$

where $\hat{G}_{aw}(t)$ is the afternoon traffic component corresponding to the initial parameter set in $\mathscr{P}_{aw}$. As shown in Figs. 3.7 and 3.8, the analytical calculation approach has achieved excellent result and greatly reduced the estimation error of the initial parameters in comparison with the K-means clustering based approach. Then, the model built with the initial parameters serves as a suitable starting point for further parameter adjustments, with the goal of continuously reducing the gap and ultimately arriving at the model parameters that yield the best performance.

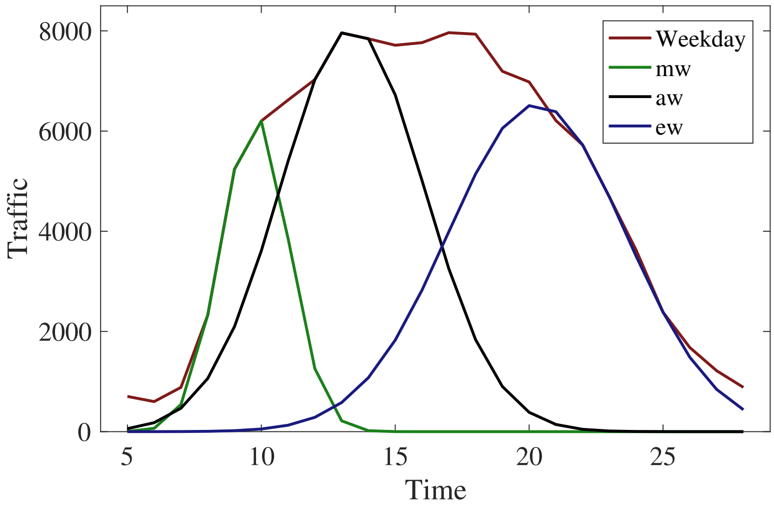


Fig. 3.7 The three traffic components corresponding to $\mathscr{P}_{aw}$, $\mathscr{P}_{aw}$, and $\mathscr{P}_{aw}$ with the minimum E_{error}.

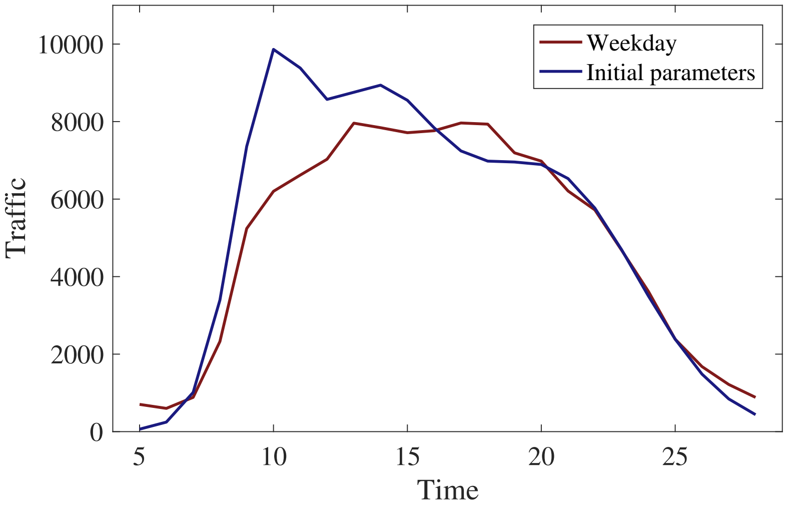


Fig. 3.8 The comparison between the real-world traffic data on weekday and the traffic profile derived from the analytical calculations approach.

3.3 Evaluation with Real-world Traffic Data

This section primarily compares the predictive accuracy and computational efficiency of the proposed UBB NTP method with benchmark methods. Predictive accuracy is measured by MSE, RMSE, MAE, and R2, which can be clearly defined by the following formulas:

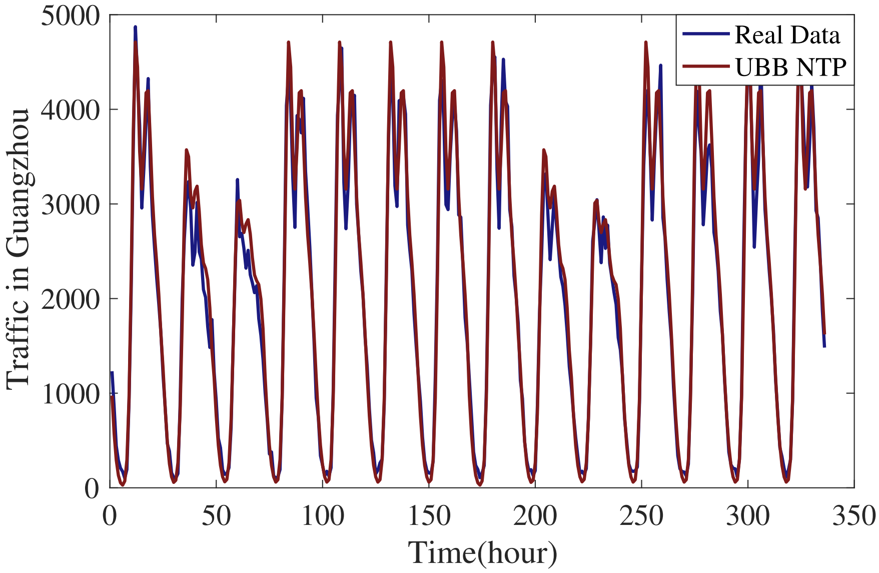


Fig. 3.9 The prediction results of UBB NTP method for Guangzhou.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.25}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \tag{3.26}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \tag{3.27}$$

$$\mathrm{R2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{3.28}$$

where n represents the number of predicted samples, y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and $\bar{y}$ denotes the mean value of y_i. Computational efficiency is represented by the elapsed time including training and prediction time.

A fully-connected neural network with five hidden layers and a LSTM network with 3 hidden layers are selected to represent ML-based methods. The ARMA model and ARIMA model are selected to represent statistics-based methods. According to the result of the BIC, the parameters are determined to be ARMA(4,2) for processing Dataset 1 and ARMA(3,1) for

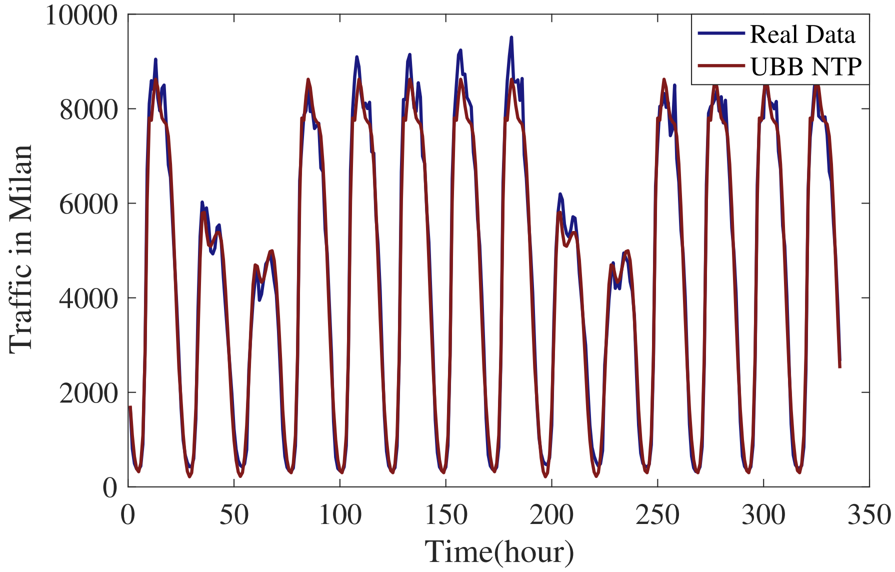


Fig. 3.10 The prediction results of UBB NTP method for Milan.

processing Dataset 2. Following the same steps, the ARIMA models are set to ARIMA(2,1,3) and ARIMA(1,1,1) for Dataset 1 and Dataset 2, respectively.

3.3.1 Performance of the UBB NTP Method

This work adopts the SMS data from the first two weeks to build the proposed mathematical model and the data from the last two weeks to evaluate its performance. As shown in Tables 3.2 and 3.3, there are two parameter sets extracted from Dataset 1 and Dataset 2, respectively. Then, the Guangzhou and Milan datasets are employed to perform the NTP task. The proposed UBB NTP method exhibits excellent performance in terms of prediction accuracy, as shown in Figs. 3.9 and 3.10. To some extent, it has verified that our assumptions regarding user behavior are realistic.

3.3.2 Comparison with Benchmark Methods

This section first compares the performance of the UBB NTP method with all benchmark methods on the Guangzhou SMS data by the metrics of MSE, RMSE, MAE and R2. As shown in Fig. 3.11, the UBB NTP method, which achieves the highest R2 and the lowest

Table 3.2 Parameters of UBB NTP method in Guangzhou.

	$R_{c,p}$	σ_c^2	$t_{c,p}$
mw	4626	3.10	12.14
aw	3839	3.91	17.35
ew	2136	6.63	22.18
msa	3612	2.89	12.09
asa	2989	3.14	16.55
esa	2356	5.78	21.60
msu	2866	2.81	11.95
asu	2759	4.13	16.47
esu	2252	6.39	22.15

Table 3.3 Parameters of UBB NTP method in Milan.

	$R_{c,p}$	σ_c^2	$t_{c,p}$
mw	2942	0.88	9.45
aw	6613	6.51	11.80
ew	7327	13.91	18.72
msa	3297	2.98	10.68
asa	3684	9.92	13.87
esa	4654	11.69	20.16
msu	3720	5.29	11.43
asu	3187	8.90	16.41
esu	3843	8.59	21.36

MSE and RMSE, slightly outperforms the LSTM network and is superior to the statistics-based methods. Fig. 3.12 shows the prediction accuracy of each method on the Milan SMS data. As shown in Fig. 3.12, the UBB NTP method and the LSTM network have almost the identical MSE, RMSE and R2. Regardless of ARMA and ARIMA models, statistics-based methods do not perform very well in terms of accuracy. Since the UBB NTP method is designed with user behaviours, it matches the practical wireless traffic demand very well.

Fig. 3.13 demonstrates the surprising superiority of the proposed UBB NTP method in terms of elapsed time. Take Dataset 1 as an example, our method completes both the training task and prediction step in only about 8.7 seconds. The efficiency is approximately 12 times the most efficient benchmark method, i.e., the ARMA model, and 28 times the most accurate benchmark method, i.e., the LSTM network. Although the multi-step mode saves some time for prediction, this time represents only a tiny fraction of the elapsed time. The key reason for the efficiency improvement is the reduced training time. Compared with benchmark methods, there is no need for offline training in the proposed UBB NTP method.

Overall, the proposed UBB NTP method obtains the best overall performance in both Datasets 1 and 2, which means this method is well adapted to traffic data from different regions.

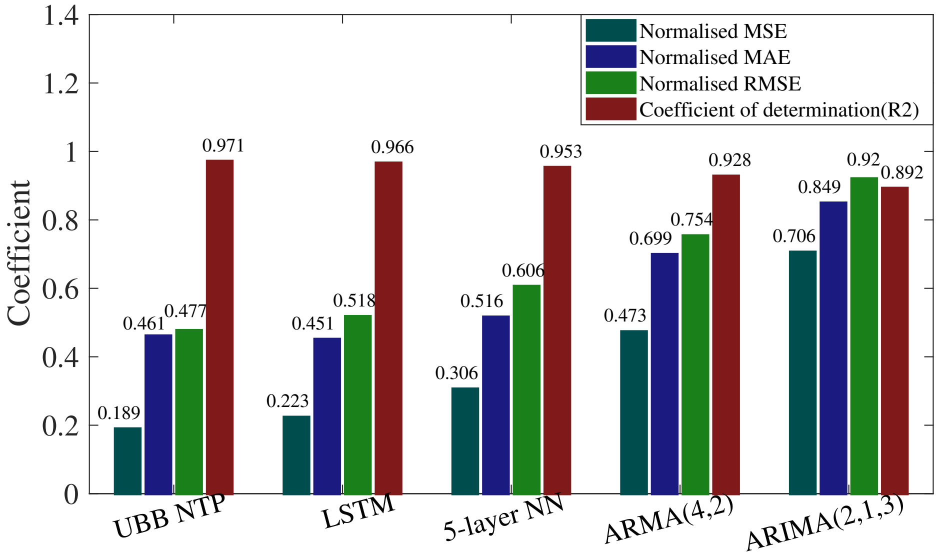


Fig. 3.11 The prediction performance of the proposed UBB NTP method and the benchmark methods for Guangzhou SMS dataset.

3.3.3 Analyses and Discussion

It is worth noting that the most significant advantage of the UBB NTP method is interpretability. It could be a link between traffic patterns and the field of social sciences. These parameters imply users' habits of using network traffic. In Tables 3.2 and 3.3, $R_{\mathrm{c,p}}$ denotes the peak of the traffic component, while $t_{\mathrm{c,p}}$ denotes the corresponding time in a 24-hour format, measured in hours. Take the parameter set of Guangzhou as an example. According to the 3-sigma rule in the normal distribution, 68% of morning traffic occurs between 10:15 am and 1:55 pm, 68% of traffic demand in every afternoon occurs between 2:30 pm and 7:20 pm, and 68% of traffic demand in every evening occurs from 7:10 pm to 0:45 am the next day. On both weekdays and weekends, the morning has the highest traffic demand, followed by the afternoon, and the evening has the lowest.

Moreover, σ_c^2 indicates the magnitude of traffic deviation; the higher the σ_c^2, the more dispersed the distribution. In both Tables 3.2 and 3.3, we can observe that σ_c^2 at night are higher than those during the day. It is mainly caused by the changes in users' state. As the day progresses, users gradually change from a working state to a leisure state. After that, users generate traffic whenever and wherever they want just for their individual needs, such

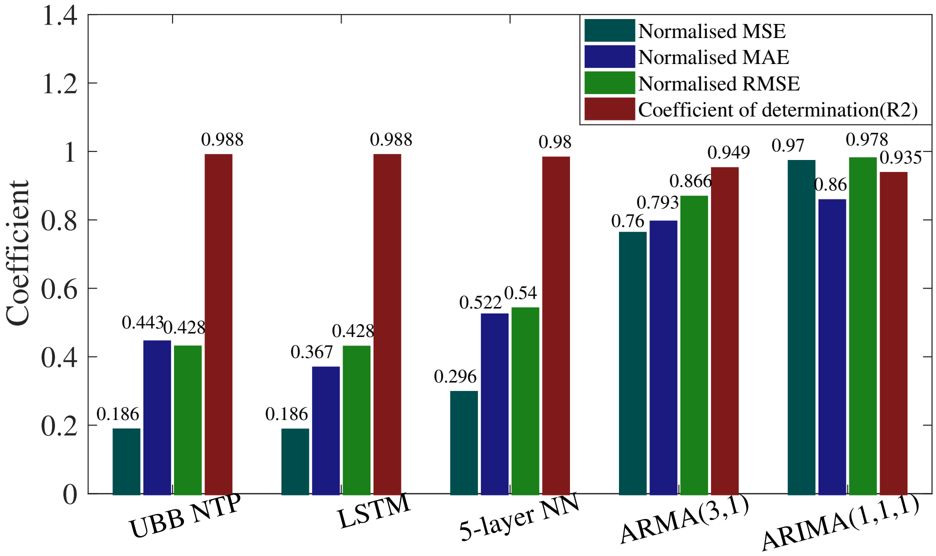


Fig. 3.12 The prediction performance of the proposed UBB NTP method and the benchmark methods for Milan SMS dataset.

as chatting, ordering take-out, etc. The distribution is, therefore, more dispersed at night. As shown in Table 3.3, σ_{ew}^2 and $R_{ew,p}$ means a large number of users are active during the period. What's more, the leisure users dominate.

By comparing Milan and Guangzhou, the traffic curves in the two cities in Figs. 3.9 and 3.10 are comparable. To some extent, it verifies that urban users have similar living habits. Furthermore, both Milan and Guangzhou have highly developed tertiary industries [89][90], which could be one of the main reasons for the similar traffic curves. On the other hand, Tables 3.2 and 3.3 demonstrate that users in Guangzhou and Milan have various preferences. The distribution of peak times of the three traffic components is more even in Guangzhou, and they are all spaced about five hours apart. While in Milan, the first two traffic components are close in time and farther apart from the one representing evening.

3.4 Conclusion

In this chapter, a novel UBB NTP method has been proposed, which exhibits higher overall performance when compared to existing machine-learning-based and statistics-based meth-

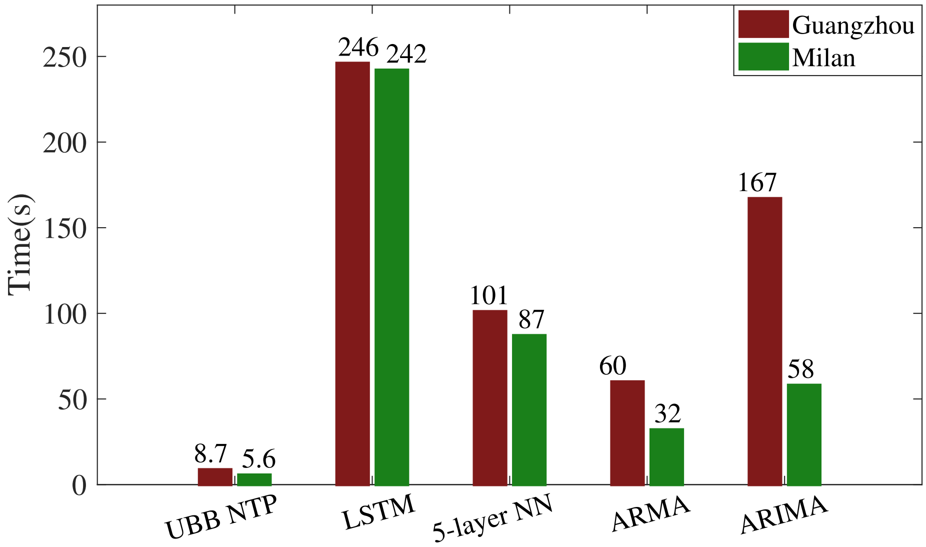


Fig. 3.13 The elapsed time of the proposed UBB NTP method and the benchmark methods.

ods. The method's parameters are concise, possess practical significance, and provide an interpretable NTP solution. Furthermore, the standardized parameter set enables comparing traffic patterns across different regions. Hence, the proposed UBB NTP method may be considered a promising aspect in the combination of communication and social science.

In the next chapter, we will focus on the nonroutine traffic caused by the nonroutine events. Based on analyzing the traffic pattern, variation trend and corresponding user behavior, we will propose the NNTP method for nonroutine traffic in the next chapter, which is an extension of the UBB NTP method in a particular context. Its prediction performance will be validated through a case study.

Chapter 4

Interpretable Nonroutine Network Traffic Prediction with a Case Study

Overview

This work pioneers a NNTP method to prospectively provide a theoretical basis for avoiding large-scale network disruption by accurate prediction of bursty traffic. Certain events impacting user behavior subsequently trigger the nonroutine traffic, which would significantly constrain the performance of NTP model. By analyzing nonroutine traffic and the corresponding events, the NNTP method is pioneered to construct interpretable NTP model. Based on the real-world traffic data, the network traffic generated during soccer games serves as a case study to validate the performance of the NNTP method. The numerical results indicate that our prediction closely fits the traffic pattern. In comparison to existing researches, the NNTP method is at the forefront of finding a balance among interpretability, accuracy, and computational complexity.

4.1 Introduction

According to analyzing and modeling of user behaviour, the previous chapter proposed the UBB NTP method, which highly matches with the real-world network traffic data aggregated in certain regions like Milan city. However, user behavior not only contains regular daily behavior but also can be impacted by some special events. This work discovers the phenomenon that certain events can bring about significant changes in cellular network traffic by impacting user behavior. This kind of traffic, referred to as nonroutine traffic, poses a great challenge to the state-of-the-art NTP models in terms of prediction accuracy, computational efficiency, and interpretability. This chapter attempts to construct a NNTP method to solve the problem along the trajectory of analyzing and modeling of user behaviour.

It is worth noting that the problem caused by nonroutine traffic, overlooked in academia, carries significant potential consequences. Specifically, the degradation in NTP performance is anticipated to result in a decline in QoS, consequently leading to a deterioration in customer satisfaction [91] and the reputation of the operator [92]. Ultimately, this may culminate in severe repercussions, including compromised future profitability [92].

The state-of-the-art NTP models exhibit inherent limitations like prediction accuracy of statistics-based models, interpretability of machine-learning-based models and so on. In addition, experimental results in this chapter indicate that these limitations are exacerbated when it encounters nonroutine traffic data. The underlying cause lies in the inability of these models to take user behaviour into consideration [11]. Specifically, the statistics-based and shallow-learning-based models have limited learning capacity [16]. The nonroutine traffic data will further hinder their performance. Although the deep-learning-based model could understand the complex temporal-spatial correlations [80], it is still difficult to extract the traffic pattern accurately when confronted with nonroutine traffic data. It is because the nonroutine data only takes up a relatively small proportion of the overall data. Meanwhile, it requires not only a large amount of nonroutine traffic data, but also an increase in the parameters and complexity of the model. This can further increase the difficulty of hyper-parameters' selection and the risk of overfitting, as well as introduce longer computation time. Moreover, ML-based models have poor interpretability for nonroutine traffic data, because its parameters do not have practical significance [11]. Consequently, the state-of-the-art NTP models perform poorly in the presence of nonroutine traffic data.

To achieve a leap from 0 to 1 in the context of nonroutine network traffic, the work in this chapter pioneers a novel NNTP method. Specifically, using the the real-world traffic data generated during soccer games as a case study, this work initially analyzes the underlying causes of the nonroutine traffic. It subsequently reveals the correlation between user behavior and traffic pattern. Finally, it formulates a dedicated model, referred to as the SG-NNTP model, for such nonroutine events. The model constructed based on the NNTP method is analytical and interpretable. In addition, numerical results show that the NNTP method performs excellently in prediction accuracy and computational efficiency.

The main contributions of this chapter are summarized as follows:

1) This work takes the lead in systematically analyzing and researching nonroutine traffic, i.e. network traffic caused by nonroutine event which differs significantly from regular traffic patterns.

2) Based on the analysis of nonroutine traffic, this work pioneers the NNTP method to construct the NTP model. Compared with the existing works, the proposed method is specifically designed for nonroutine traffic, takes into account the impact of the nonroutine events on the wireless network traffic patterns, and utilizes the information of nonroutine events to construct the NTP model, which achieves the optimal prediction accuracy, computational efficiency, and interpretability. The NNTP method is an important inspiration for future research on nonroutine traffic.

3) Following the NNTP method, this work formulates the SG-NNTP model as a case study. Compared with benchmark models which do not take nonroutine traffic into consideration, the NNTP method improves the prediction accuracy, both in multi-step and single-step prediction mode. What is more, the NNTP method decreases the elapsed time and improves the computational efficiency a lot. In addition, the NNTP method has outstanding interpretability and is easy to migrate to similar situations.

The rest of the chapter is structured as follows. Section 4.2 offers the definition and categorization of nonroutine traffic. It subsequently introduces the key points of the NNTP method and formulates the SG-NNTP model as a case study. In addition, Subsection 4.2.5 and 4.2.6 discuss the multi-step and single-step prediction mode, respectively. In Section 4.3, the predictions of the SG-NNTP model and benchmark models are performed, and the performance of these models is evaluated in both multi-step and single-step prediction mode. Finally, this chapter concludes this work in Section 4.4.

4.2 The Proposed NNTP Method with a Case Study

This section proposes the NNTP method to enhance the performance of the NTP model in the presence of nonroutine traffic data. Following this method, we formulate the SG-NNTP model with the real-world traffic data gathered during soccer games.

4.2.1 Analysis of Nonroutine Traffic

In a specific geographical area, the daily activities of local users typically exhibit a high degree of cyclical and repetitive pattern. The network traffic that determined by user behavior demonstrates a similar variation trend. Therefore, the UBB NTP model can quickly and accurately capture the daily traffic pattern [11].

However, the incidence of nonroutine events in the region may influence user behavior or the quantity of user, subsequently exerting a substantial impact on the traffic pattern [93]. When the region hosts significant events, such as sports games and concerts, it tends to draw a considerable influx of short-term users. In this context, short-term users refer to individuals who arrive in this region specifically for the event and stay there during its duration. The traffic generated by these short-term users is significantly different from the daily traffic. This type of nonroutine traffic is defined as additive nonroutine traffic. As the name suggests, in this case, the overall traffic in the area can be regarded as a superposition of the daily traffic generated by resident users and the nonroutine traffic generated by short-term users.

Similarly the quantity of resident users or the overall user habits in the region is also expected to change due to certain factors, which in turn leads to other kind of nonroutine network traffic rather than the additive nonroutine traffic. The causes of these situations are more complex and difficult to collect data, and will be the direction of our future research. Following a progression from simplicity to complexity, this chapter focuses on the processing of additive nonroutine traffic.

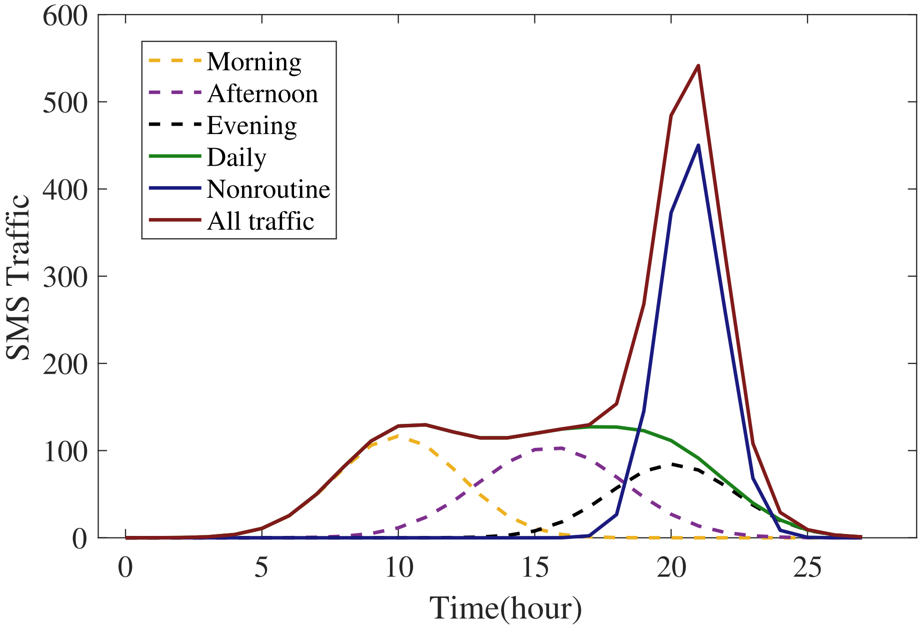


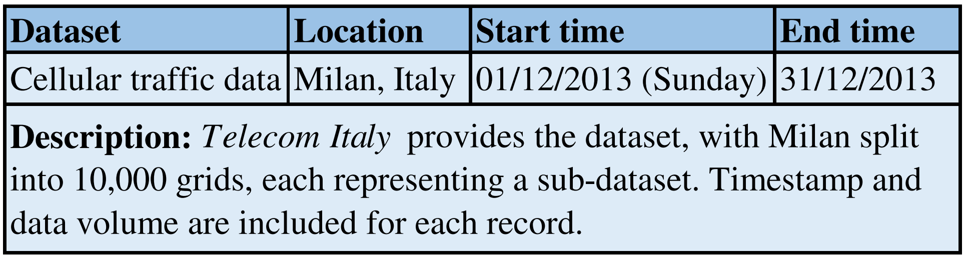
Fig. 4.1 The daily traffic and the additive nonroutine traffic.

4.2.2 The Proposed NNTP Method

Based on the analysis of nonroutine traffic, this chapter proposes the NNTP method. In simple terms, the NNTP method involves analyzing and summarizing the traffic data corresponding to the nonroutine events, and formulating specific NTP models for similar events.

Many events in the daily lives of users could make the daily traffic pattern change abnormally. The occurrences of these events are often scheduled rather than completely random and unexpected, such as fairs, ball games, concerts, carnivals, and so on. Some of the information related to these scheduled events is available in advance, referred to as advanced information, such as the commencement time, the event duration, the expected attendance or ticket sales, and the type of the event. Relying solely on historical traffic data to acquire these advanced information is difficult and demanding.

While, it would be a shortcut to directly utilize these easily accessible advanced information to construct NTP models corresponding to these nonroutine events. Meanwhile, with today's data explosion, effective multi-source data application is becoming an essential research. Hence, the efficient utilization of multi-source data is also one of the innovations



Dataset	Location	Start time	End time
Cellular traffic data	Milan, Italy	01/12/2013 (Sunday)	31/12/2013
Description: *Telecom Italy* provides the dataset, with Milan split into 10,000 grids, each representing a sub-dataset. Timestamp and data volume are included for each record.			

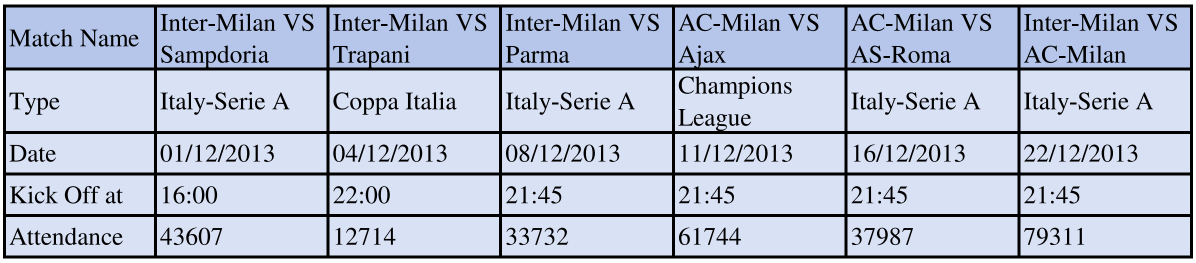
Fig. 4.2 Cellular network traffic data in Milan published by Telecom Italia.

of the NNTP method. Specifically, this chapter decomposes the total traffic during the nonroutine event into a superposition of the daily traffic and the additive nonroutine traffic. As shown in Fig. 4.1, the solid green line depicts the daily traffic, the solid blue one depicts the additive nonroutine traffic, and the solid red one represents the total traffic. It is worth noting that the horizontal axis in Fig. 4.1 represents time in hours. The values of the horizontal axis should not exceed 23; any values greater than or equal to 24 should be attributed to the next day. However, for the sake of axis continuity, a representation beyond 23 has been adopted in this chapter.

Algorithm 1: the Proposed NNTP Method

```
Input: $\mathbf{S}$, $\mathbf{AI} = [\mathbf{h}, \mathbf{d}, \mathbf{g}, \mathbf{m}, \ldots]$
Dividing the historical traffic data $\mathbf{S}$ into two parts: $\mathbf{S}_\alpha$ and $\mathbf{S}_\beta$
Construct the Module.UBB NTP based on $\mathbf{S}_\alpha$
$\mathbf{S}_{\beta,\text{daily}}$ = Module.UBB NTP$(\mathbf{h}, \mathbf{d})$ % Predict the daily traffic component in $\mathbf{S}_\beta$
$\mathbf{S}_{\beta,\text{nonroutine}} = \mathbf{S}_\beta - \mathbf{S}_{\beta,\text{daily}}$
for $i = 1$ : lenth$(\mathbf{AI})$ do
    Get $\mathbf{AI}[i]$
    Infer the connection $\mathbf{C}[i]$ between $\mathbf{S}_{\beta,\text{nonroutine}}$ and $\mathbf{AI}[i]$
end for
Output: Construct the specific NNTP model based on $\mathbf{C}$ and $\mathbf{S}_{\beta,\text{nonroutine}}$
```

Next, the pseudo-code **Algorithm 1** is used for introducing the proposed NNTP method. In **Algorithm 1**, $\mathbf{S}$ represents the historical traffic data. $\mathbf{S}$ is categorized into two groups, i.e. $\mathbf{S}_\alpha$ and $\mathbf{S}_\beta$, depending on the occurrence of nonroutine events. $\mathbf{S}_\alpha$ represents the total traffic data generated on days when the nonroutine events do not occur, while $\mathbf{S}_\beta$ represents the total traffic data generated on days when the nonroutine events take place. Then, we construct the UBB NTP model that relies on $\mathbf{S}_\alpha$ and forecast the the daily traffic component $\mathbf{S}_{\beta,\text{daily}}$ in $\mathbf{S}_\beta$.



Match Name	Inter-Milan VS Sampdoria	Inter-Milan VS Trapani	Inter-Milan VS Parma	AC-Milan VS Ajax	AC-Milan VS AS-Roma	Inter-Milan VS AC-Milan
Type	Italy-Serie A	Coppa Italia	Italy-Serie A	Champions League	Italy-Serie A	Italy-Serie A
Date	01/12/2013	04/12/2013	08/12/2013	11/12/2013	16/12/2013	22/12/2013
Kick Off at	16:00	22:00	21:45	21:45	21:45	21:45
Attendance	43607	12714	33732	61744	37987	79311

Fig. 4.3 The information of the soccer games hosted by the San Siro Stadium in December 2013.

Thus, the nonroutine traffic component $\mathbf{S}_{\beta,\text{nonroutine}}$ in $\mathbf{S}_{\beta}$ can be obtained. **AI** represents the advanced information, including but not limited to the commencement time **h**, the duration **d**, the type **g**, and the attendance **m** of the nonroutine events. By conducting a thorough analysis of nonroutine traffic, we can extract the relationship **C** between $\mathbf{S}_{\beta,\text{nonroutine}}$ and **AI**. Finally, the specific NNTP model can be constructed. Next, this chapter will detail the process through a case study.

4.2.3 Dataset for Nonroutine Traffic

This chapter adopts the real-world network traffic data, as shown in Fig. 4.2, published by Telecom Italia, a large European telecommunications service operator [88]. In the spatial dimension, Milan city is covered by 10,000 grids of size 235 × 235 meters. Each data point within every grid represents the cellular traffic data generated by local users during the time interval between two consecutive timestamps.

By searching the grids surrounding the G.MEAZZA SAN SIRO, we obtained traffic data in the vicinity of the San Siro stadium. The time granularity is set at one hour, which means that there are 24 data samples each day. Then, this chapter researches the information of the soccer game hosted by the San Siro Stadium in December 2013, as shown in Fig. 4.3.

Authors in [94] demonstrates that the traffic data, including Short Message Service (SMS), calls, internet, etc., is directly related to the soccer games and contains similar nonroutine traffic pattern. Take SMS data as an example, Fig. 4.4 plots the curves of the traffic data corresponding to all of the soccer games in Fig. 4.3. Visual inspection reveals an alignment

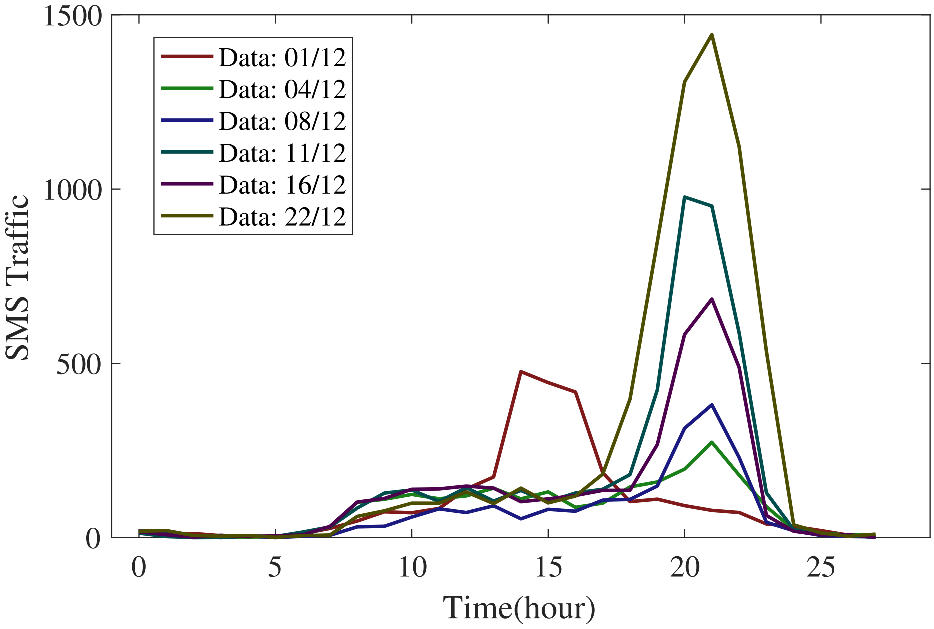


Fig. 4.4 SMS data for the soccer games held at the San Siro stadium in December.

between the time of peaks and the period of the soccer game, which is consistent with the conclusion given by F. Botta *et al.* in [94].

4.2.4 The Case Study: SG-NNTP Model

Based on the above analysis, the traffic brought by soccer games belongs to the additive nonroutine traffic. To extract the nonroutine component, the initial step is to obtain the daily traffic component with the UBB NTP method. There are three dedicated daily traffic models designed for weekday, Saturday, and Sunday, respectively [11]. As shown in Fig. 4.1, any of the daily traffic model represented by solid green line is a superposition of yellow, purple, and black dashed lines which represent the traffic components with the morning, afternoon and nighttime traffic as the main body, respectively [11]. Therefore a total of nine traffic components are required for a whole week. The abbreviations of these traffic components are listed in Table 4.1. Each traffic component is modeled as a Gaussian signal that can be expressed as

$$G_{\mathrm{c}}(t) = R_{\mathrm{c}}\exp\left(-\frac{(t-t_{\mathrm{c}})^2}{2\sigma_{\mathrm{c}}^2}\right), \tag{4.1}$$

Table 4.1 The symbols corresponding to the 9 traffic components.

	Weekday	Saturday	Sunday
Morning	mw	msa	msu
Afternoon	aw	asa	asu
Evening	ew	esa	esu

where t is in a 24-hour format, t_{c} and σ_{c}^2 denote the mean value and variance of the Gaussian signal, respectively. R_{c} is the peak value of the traffic component. $\mathrm{c} \in \{\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3\}$, represent weekday, Saturday and Sunday, respectively, and $\mathrm{c}_1 \in \{\mathrm{mw}, \mathrm{aw}, \mathrm{ew}\}$, $\mathrm{c}_2 \in \{\mathrm{msa}, \mathrm{asa}, \mathrm{esa}\}$, $\mathrm{c}_3 \in \{\mathrm{msu}, \mathrm{asu}, \mathrm{esu}\}$. Therefore, the hourly traffic at time t, the kth day of a week, can be represented as

$$
\begin{aligned}
Y_k(t) = &\sum_{\mathrm{c}_1} R_{\mathrm{c}_1} \sum_{n_{\mathrm{d}}=1}^{5} \exp\left(-\frac{\left(t+24(k-n_{\mathrm{d}})-t_{\mathrm{c}_1}\right)^2}{2\sigma_{\mathrm{c}_1}^2}\right) \\
&+\sum_{\mathrm{c}_2} R_{\mathrm{c}_2} \exp\left(-\frac{\left(t+24(k-6)-t_{\mathrm{c}_2}\right)^2}{2\sigma_{\mathrm{c}_2}^2}\right) \\
&+\sum_{\mathrm{c}_3} R_{\mathrm{c}_3} \exp\left(-\frac{\left(t+24(k-7)-t_{\mathrm{c}_3}\right)^2}{2\sigma_{\mathrm{c}_3}^2}\right).
\end{aligned}
\tag{4.2}
$$

with the index of the day $k \in \{1,2,3,4,5,6,7\}$, the index of the weekday n_{d}. The optimal parameter set of the daily traffic can be obtained by optimizing the following equation:

$$
\underset{\substack{R_{\mathrm{c}}, t_{\mathrm{c}}, \sigma_{\mathrm{c}}^2 \\ \mathrm{c} \in \left\{\substack{\mathrm{mw,aw,ew,} \\ \mathrm{msa,asa,esa,} \\ \mathrm{msu,asu,esu}}\right\}}}{\text{minimise}} \quad \sum_{k=1}^{7} \sum_{t=1}^{24} \| Y_k(t) - Y_{\mathrm{measure}}(t) \|^2, \tag{4.3}
$$

where $Y_{\mathrm{measure}}(t)$ refers to the traffic measurement at the moment t. This chapter uses the gradient descent approach to tackle the minimization problem.

According to the observation of Fig. 4.4, it is obvious that the practical traffic data caused by the soccer games are similar to a bell-shaped curve. We assume that the behavior of attendances using cellular network services obeys an independent identically distribution, which is a logical general assumption. Then according to the central limit theorem, the prior distribution of the additive nonroutine traffic is naturally constructed as a Gaussian signal, which is consistent with the observation. Hence, the the additive nonroutine traffic can be represented as

$$Y_{\text{additive}}(t) = P\exp\left(-\frac{\left(t-t_{\text{sg}}\right)^2}{2\sigma_{\text{sg}}^2}\right), \tag{4.4}$$

$$P = \frac{R_{\text{sg}}}{\sigma_{\text{sg}}\sqrt{2\pi}}, \tag{4.5}$$

where the granularity of time t is an hour. P is the peak value of the nonroutine traffic. R_{sg} represent the amplitude parameter. t_{sg} and σ_{sg}^2 are the mean and the variance of the Gaussian signal, respectively. As shown in Fig. 4.4, the horizontal coordinates of the peaks have a strong connection with the commencement time of the soccer games.

The traffic volume generated on the day when the soccer game occurs is equal to the daily component plus the additive nonroutine component, which is given by

$$Y_{\text{total}}(t) = Y_{\text{daily}}(t) + Y_{\text{object}}(t), \tag{4.6}$$

where $Y_{\text{object}}(t)$ is ideal for the additive nonroutine component at the moment t. Since the daily traffic component $Y_{\text{daily}}(t)$ can obtained by UBB NTP method as mentioned above, $Y_{\text{object}}(t)$ can be obtained according to Equation 4.6. The least squares method is then used to minimize the difference between Y_{object} and Y_{additive}, which is formulated as

$$\underset{R_{\text{sg}},t_{\text{sg}},\sigma_{\text{sg}}^2}{\text{minimise}} \sum ||Y_{\text{additive}}(t) - Y_{\text{object}}(t)||^2, \tag{4.7}$$

and thus, the parameters, i.e. t_{sg}, R_{sg}, and σ_{sg} corresponding to each soccer game, are obtained.

Figs. 4.5, 4.6, 4.7, and 4.8 illustrate the fitting performance of the SG-NNTP model for the first 4 soccer games in December 2013 at San Siro stadium, Milan. Approximating t_{sg} to the commencement time of the soccer games simplified the problem and led to successful outcomes. The parameters of the simulation are shown in Table 4.2.

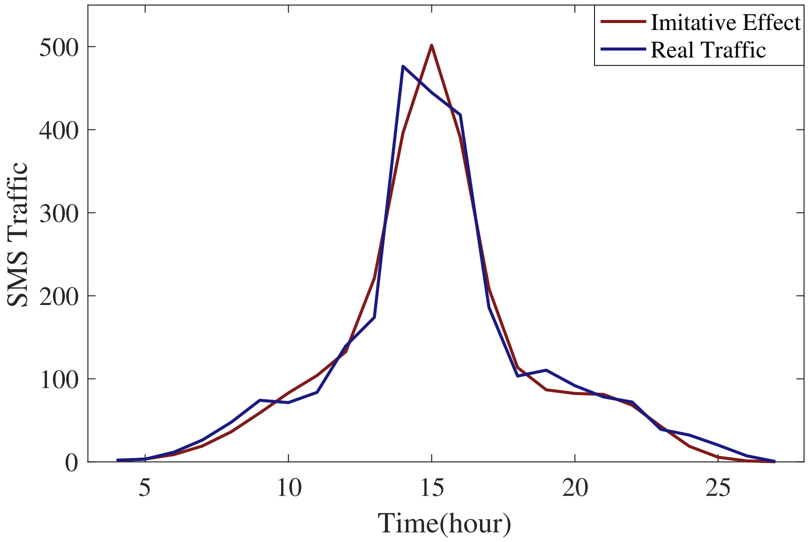


Fig. 4.5 The imitative effect of SG-NNTP model for the soccer game: Inter-Milan vs. Sampdoria kicks off at 16:00 December 1, 2013.

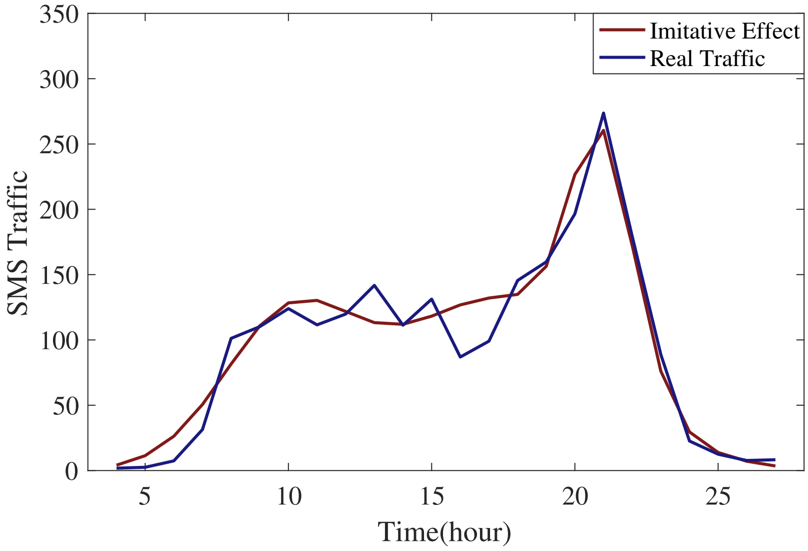


Fig. 4.6 The imitative effect of SG-NNTP model for the soccer game: Inter-Milan vs. Trapani kicks off at 22:00 December 4, 2013.

4.2.5 Multi-step Prediction of SG-NNTP

One of the major advantages of the analytical model is that it can efficiently and accurately perform multi-step predictions. It requires the ability to estimate the initial parameters of the SG-NNTP model in advance based on the advanced information of the event. The first four

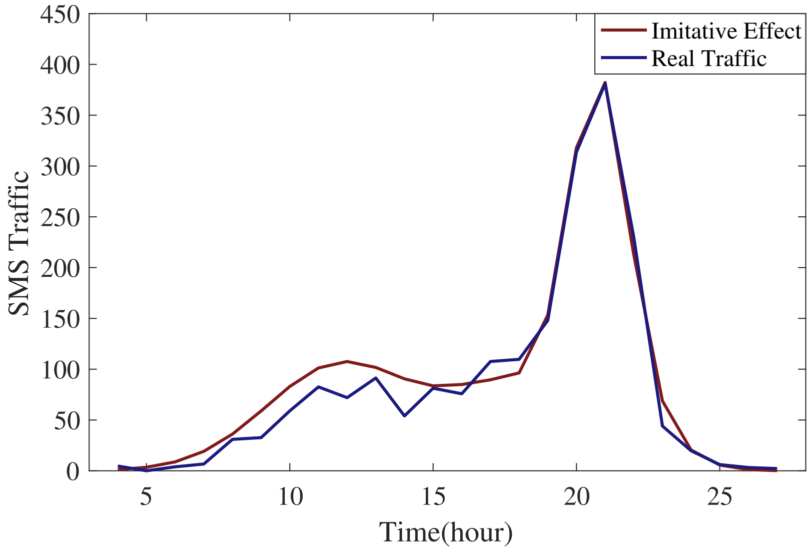


Fig. 4.7 The imitative effect of SG-NNTP model for the soccer game: Inter-Milan vs. Parma kicks off at 21:45 December 8, 2013.

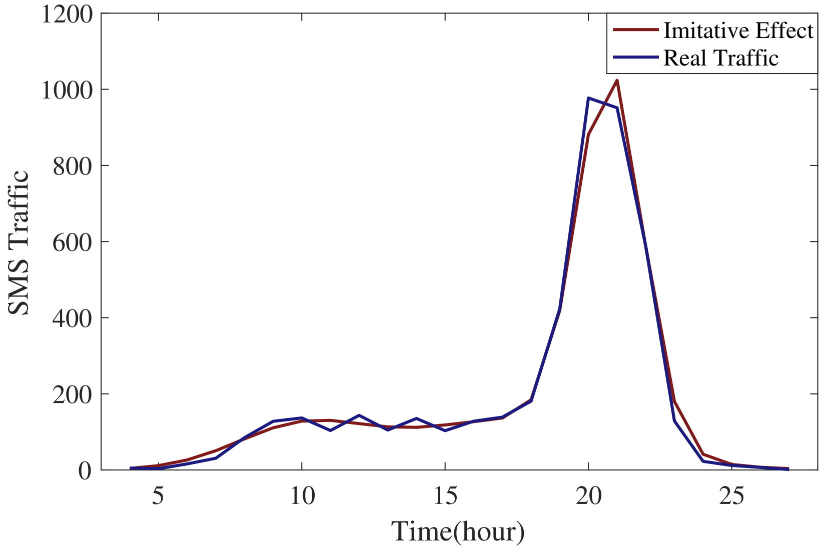


Fig. 4.8 The imitative effect of SG-NNTP model for the soccer game: AC-Milan vs. Ajex kicks off at 21:45 December 11, 2013.

soccer games hosted by the San Siro Stadium in December 2013 are then used as a training set in this chapter to derive the relationships between attendance and the initial parameters.

Table 4.2 The parameters corresponding to the first 4 soccer games in December.

Date	Dec. 01	Dec. 04	Dec. 08	Dec. 11
t_{sg}	15	21	20.75	20.75
σ_{sg}	1.263	1.176	1.011	1.155
R_{sg}	829.8	349.8	769.7	2059.8

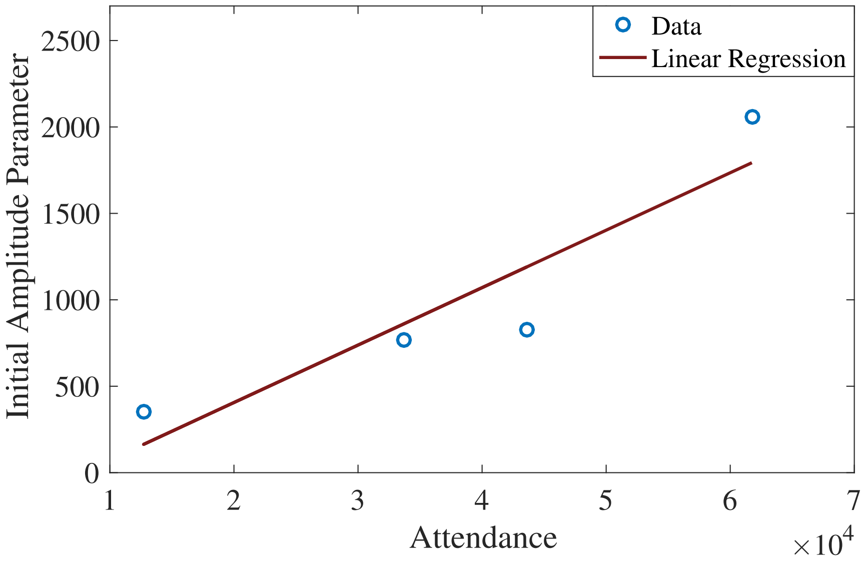


Fig. 4.9 Linear regression between R_{sg} and attendance.

The data volume of additive nonroutine traffic can be expressed as the integral of Y_{additive} over $(-\infty,+\infty)$, exactly as R_{sg}, which is given by

$$\int_{-\infty}^{+\infty} \frac{R_{\mathrm{sg}}}{\sigma_{\mathrm{sg}}\sqrt{2\pi}} \exp\left(-\frac{\left(t-t_{\mathrm{sg}}\right)^2}{2\sigma_{\mathrm{sg}}^2}\right) = R_{\mathrm{sg}}. \tag{4.8}$$

The Pearson correlation coefficient (r) between initial parameter R_{sg_j} and attendance A_j is calculated as

$$r = \frac{\sum_{j=1}^{n}(R_{\mathrm{sg}_j}-\bar{R})(A_j-\bar{A})}{\sqrt{\sum_{j=1}^{n}(R_{\mathrm{sg}_j}-\bar{R})^2}\cdot\sqrt{\sum_{j=1}^{n}(A_j-\bar{A})^2}}, \tag{4.9}$$

where $n=4$ denotes the four soccer games, R_{sg_j} and A_j denote the attendance and initial amplitude parameters corresponding to the jth soccer game, respectively. After calculation, the value of r is 92.2%, which means there is a strong positive correlation between R_{sg} and attendance. With a sample size of only 4, there is little benefit in constructing complicated correspondence between R_{sg} and attendance. Hence, this chapter applies the simplest linear regression approach to determine the correspondence as shown in Fig. 4.9. The correspondence is expressed as

$$R_{\text{sg}_j} = A_j \times 0.03323 - 258.85. \tag{4.10}$$

The parameter σ_{sg} characterizes the habits of all attendance in using cellular network traffic during the soccer game period. σ_{sg} has a complex relationship with external information such as attendance, type of game, the popularity, the intensity, etc. The implicit relationship between σ_{sg} and external information cannot be fully corroborated due to the small amount of samples. Therefore, this chapter adopts the average value to represent σ_{sg} of SG-NNTP model. With the ability to estimate the initial parameters, consequently our model is able to perform multi-step prediction.

4.2.6 Single-step Prediction of SG-NNTP

The single-step mode predicts the forthcoming traffic value by utilizing the actual traffic data at the present moment. Compared to multi-step prediction that focuses on the overall trend, single-step prediction is only concerned with the traffic value at the next moment. With the support of real time data entered at each time step, single-step prediction produces more accurate results.

The proposed SG-NNTP model also possesses the capability to execute single-step prediction, thereby achieving outstanding performance. The parameters of the model are constantly updated based on real-time practical input data. We adopt the least squares method to minimize the following equation to achieve the update of the parameters, which is formulated as

$$\underset{R_{\text{sg}_i},\, \sigma^2_{\text{sg}_i}}{\text{minimise}} \sum_{t_1}^{t_i} \| Y_{additive}(t_i) - Y_{\text{object}}(t_i) \|^2. \tag{4.11}$$

Then R_{sg_i} and σ_{sg_i} are used to predict the traffic at the moment t_{i+1}, where the predicted values at the initial moment are obtained from the initial parameters.

In addition, to reduce the impact of initial parameters on the prediction performance, a least squares optimization method with multiple initial values has been adopted. In each step of the update, multiple sets of initial parameters are selected for optimization and the smallest MSE counterpart is used for predicting the next moment traffic.

4.3 Evaluation with Real-world Traffic Data

This section primarily evaluates the prediction accuracy and computational efficiency of the proposed SG-NNTP model in comparison to benchmark models. The evaluation indexes of prediction accuracy are MSE, RMSE, MAE, and R2, which are defined as follows

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{4.12}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \tag{4.13}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \tag{4.14}$$

$$\mathrm{R2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{4.15}$$

where n denotes the size of test data. y_i and $\hat{y}_i$ denote the truth values and the predicted values of the test data, respectively. $\bar{y}$ represents the mean value of y_i. The total elapsed time of the corresponding model is used to measure the computational efficiency, including the training and prediction periods.

Deep MLP network and classic recurrent neural networks (i.e. LSTM network), are chosen to construct the benchmark models which represent the ML-based NTP model. Meanwhile, the ARMA and ARIMA models are also used to represent the classic statistics-

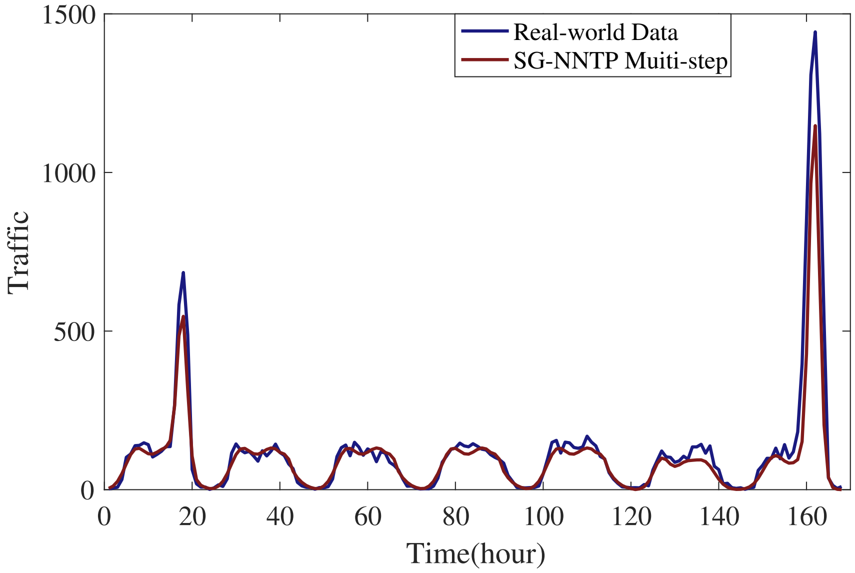


Fig. 4.10 Prediction results for SG-NNTP model in multi-step prediction mode.

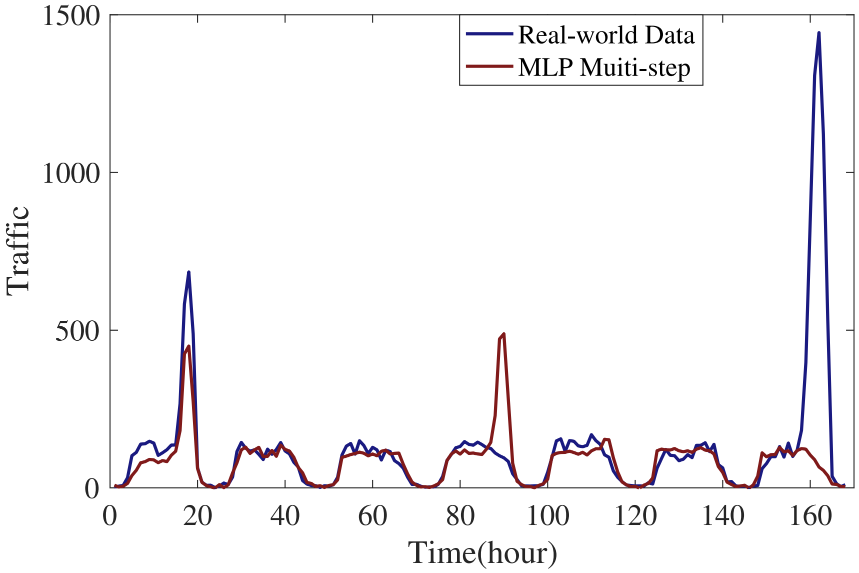


Fig. 4.11 Prediction results for the 5-Layer MLP model in multi-step prediction mode.

based NTP model. According to BIC, the parameters of ARMA and ARIMA models are determined to be ARMA (1,2) and ARIMA (3,1,1). Traffic data from December 1st to 15th is used as the training set for all models, and the data from December 16th to 22th is used as the test set. As shown in Fig. 4.3, the training set contains four pieces of additive nonroutine traffic corresponding to four soccer games and the test set contains two.

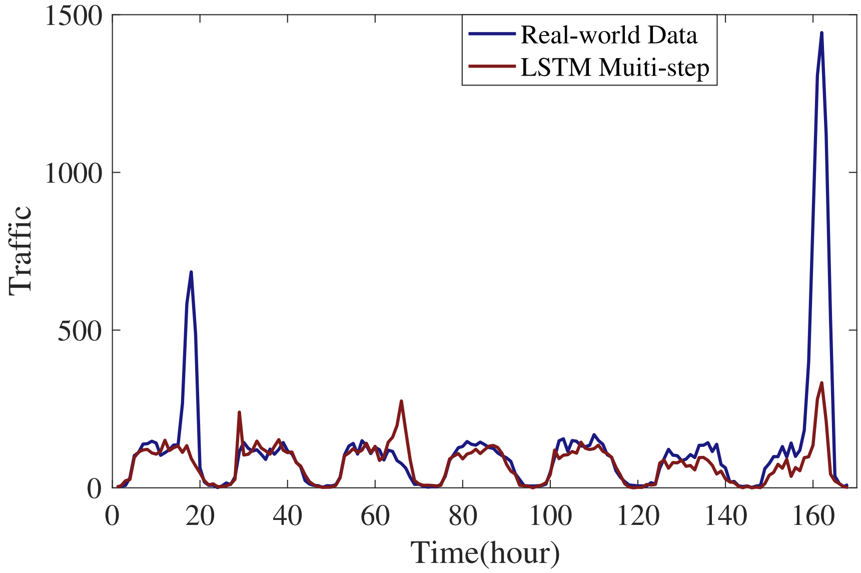


Fig. 4.12 Prediction results for the 5-Layer LSTM model in multi-step prediction mode.

4.3.1 Performance Comparison Between the SG-NNTP Model and the Benchmark Models in Multi-step Prediction Mode

The capacity to predict many time steps reflects NTP models' understanding of overall trends of cellular network traffic. The longer the processing step, the higher the level of difficulty, and the higher the likelihood that prediction accuracy will be compromised. In this chapter, the time step is set to one hour. When the prediction step size is set to 168 time steps, i.e., directly predicting the traffic for a whole week, the proposed SG-NNTP model exhibits excellent performance in terms of prediction accuracy with the R2 coefficient of 86.4%, as shown in Fig. 4.10. As benchmark models, this chapter selects 5-layer MLP network and 5-layer LSTM network to represent ML-based NTP model, and the ARMA and ARIMA models to represent statistics-based models. Experimental results demonstrate that these benchmark models can not effectively extract the traffic pattern when the prediction step size is too long. Therefore, we lower the level of difficulty by shortening the prediction step size of the benchmark models to 24 time steps. At this point, the statistics-based ARMA and ARIMA models still cannot work well. While ML-based models are superior to statistics-

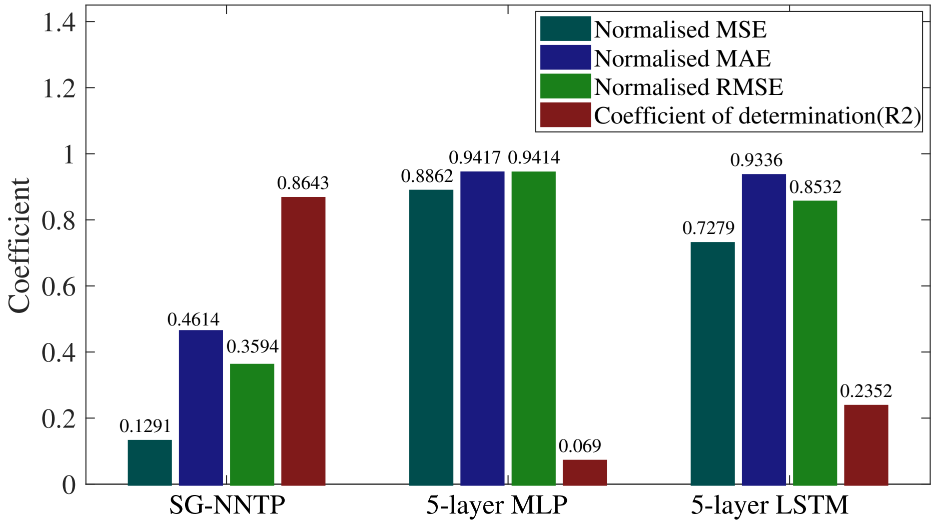


Fig. 4.13 The accuracy of the SG-NNTP model and the benchmark models in multi-step prediction mode.

based models, it is significantly inferior to the SG-NNTP model. The prediction results of the MLP and LSTM networks are shown in Figs. 4.11 and 4.12, respectively.

As can be seen from Figs. 4.10, 4.11, and 4.12 in the presence of the nonroutine event, the benchmark models can not effectively extract the traffic pattern and understand the overall trend of network traffic from the historical data. In the case of reduced prediction difficulty of benchmark models, i.e., shorter prediction step size, the benchmark models are still far inferior to the proposed SG-NNTP model, both in terms of prediction accuracy and computational efficiency. As shown in Fig. 4.13, the SG-NNTP model achieves the highest R2 which is about 3.7 to 12.5 times higher than the R2 of the benchmark models. Meanwhile, the proposed SG-NNTP model decreases the MSE, MAE, and RMAE coefficients by 82.3%, 50.6%, and 57.9%, respectively, in comparison to the LSTM models which is the best performing of the benchmark models. From the perspective of computational efficiency, as shown in Fig. 4.14, our model is 16.5 times more efficient than the 5-layer LSTM network and 23 times more efficient than the 5-layer MLP network.

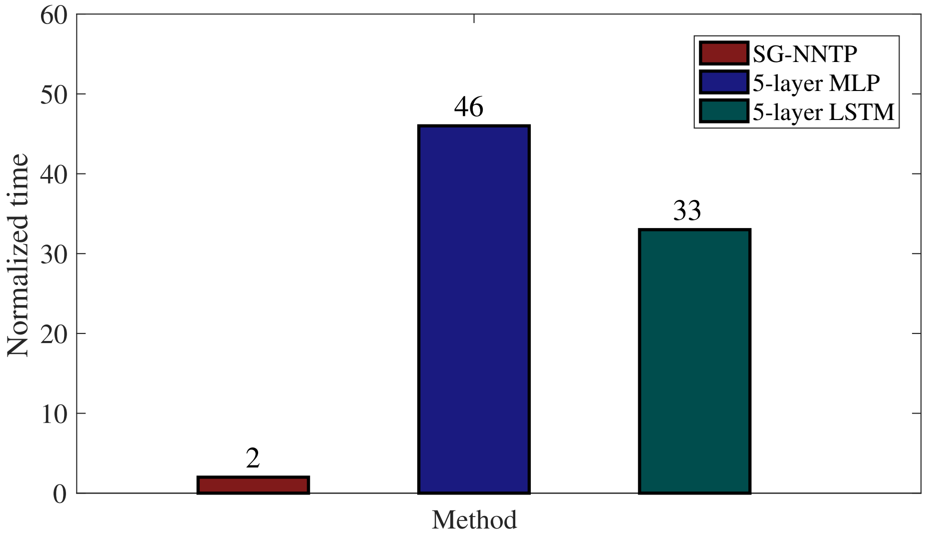


Fig. 4.14 The elapsed time of the SG-NNTP model and the benchmark models in multi-step prediction mode.

4.3.2 Performance of the SG-NNTP Model and Benchmark Models in Single-step Prediction Mode

The state-of-the-art regarding NTP mainly focuses on single-step prediction mode. This mode reflects NTP model's ability to capture localized characteristics of the traffic data. NTP models which adopt single-step prediction mode tend to have higher accuracy, due to the support of real-time traffic data at each time step. As shown in Figs. 4.15, 4.16, 4.17, 4.18, and 4.19, both the SG-NNTP model and the benchmark models show an intuitive improvement in terms of prediction accuracy when adopting the single-step prediction mode.

Figs. 4.15, 4.16, 4.17, 4.18, and 4.19 represent the prediction results of the benchmark models and the proposed SG-NNTP model. In comparison with the benchmark models, it is obviously that the proposed SG-NNTP model achieves prominent advantage in prediction accuracy as shown in Fig. 4.19. More intuitively, Fig. 4.20 quantifies their performance with the evaluation indexes of prediction accuracy. Among the benchmark models, the LSTM network achieves the best performances in terms of MSE, RMSE, and R2 coefficient, while the MLP network possesses the minimum MAE. However, there is still a considerable

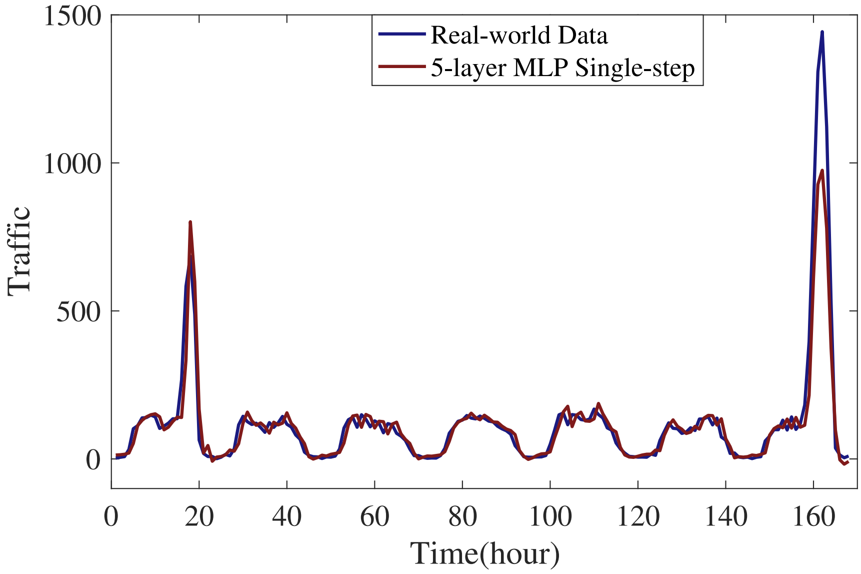


Fig. 4.15 Prediction results for the 5-layer MLP model in single-step prediction mode.

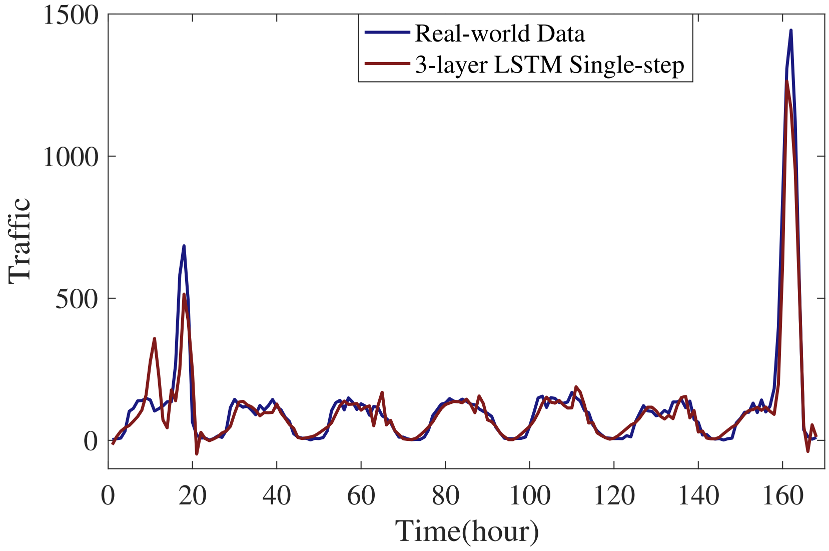


Fig. 4.16 Prediction results for the 3-layer LSTM model in single-step prediction mode.

gap between the benchmark models and the SG-NNTP model. As shown in Fig. 4.20, the SG-NNTP model achieves the R2 coefficient as high as 98.2%, while decreasing the the MSE and RMSE coefficients by 81.8% and 56%, respectively, in comparison to the LSTM network. Furthermore, the SG-NNTP model reduces the MAE coefficient by 45.3% compared with the MLP network. Fig. 4.21 demonstrates the performances of all models in terms of elapsed

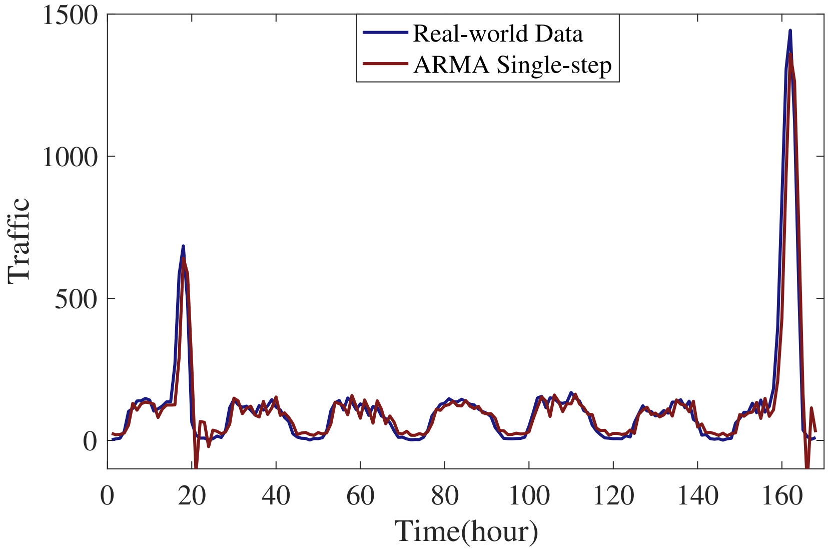


Fig. 4.17 Prediction results for the ARMA(1,2) model in single-step prediction mode.

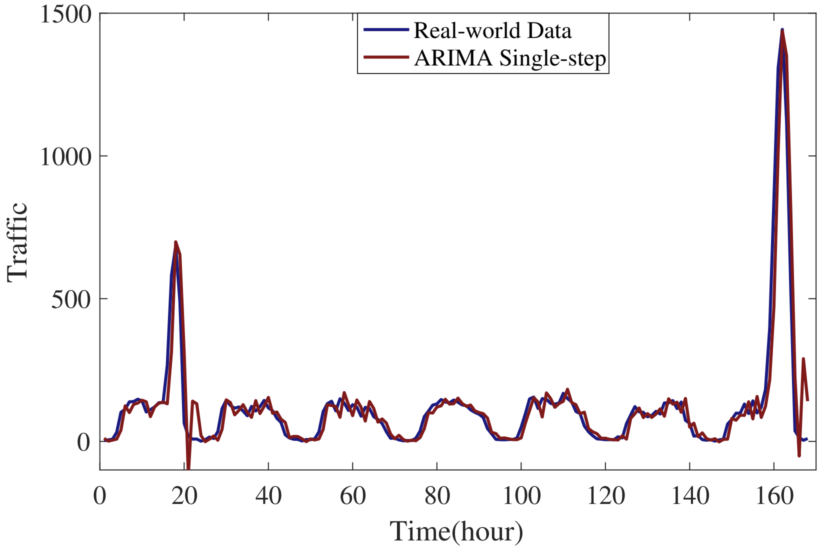


Fig. 4.18 Prediction results for the ARIMA(3,1,1) model in single-step prediction mode.

time. Our model also has the highest computational efficiency which is about 33 times that of the MLP network and about 31 times that of the LSTM network as shown in Fig. 4.21.

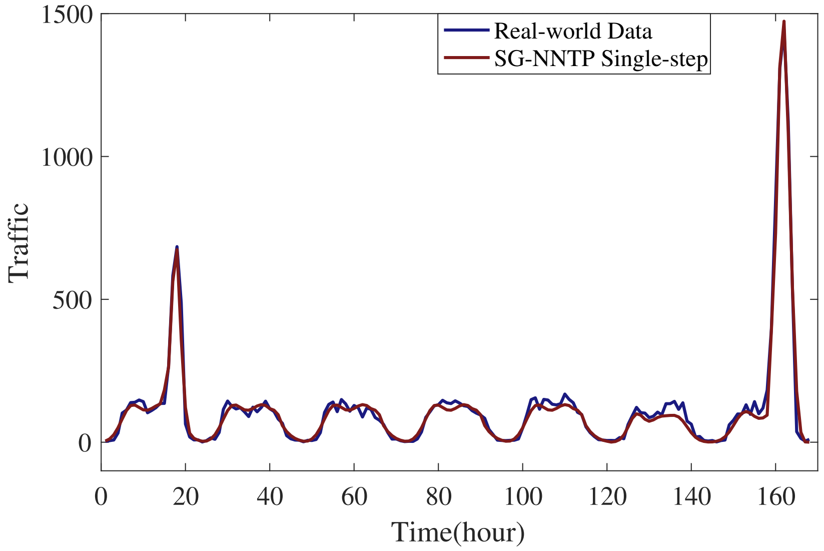


Fig. 4.19 Prediction results for the SG-NNTP model.

4.3.3 Analyses and Discussion

Due to the diversity of event types, occurrence times, durations, and user behaviors during events, NTP models built entirely on historical traffic data suffer from low prediction accuracy, computational efficiency, and interpretability. In contrast, the prediction accuracy and computational efficiency of the proposed SG-NNTP model outperform those of the benchmark models, both in single-step prediction mode and multi-step prediction mode. Meanwhile, the SG-NNTP model is an analytical NTP model constructed according to the novel NNTP method based on the analysis of user behavior. Therefore, this model possesses highly interpretability. As with pure analytical models, the model parameters are concise, and practically meaningful, which can be efficiently and accurately applied to similar events. Mutually, excellent prediction performance and generalization abilities also indirectly proves that the novel NNTP method is reasonable and consistent with network traffic pattern.

More importantly, the NNTP method pioneers the correspondence between infrequent events and specific NTP model, provides an approach to analysis and process the network traffic caused by nonroutine events, and is an inspiration for subsequent research on nonroutine traffic. If the NNTP method can be widely used, various NTP models related to different nonroutine events will be developed to construct a comprehensive database, which

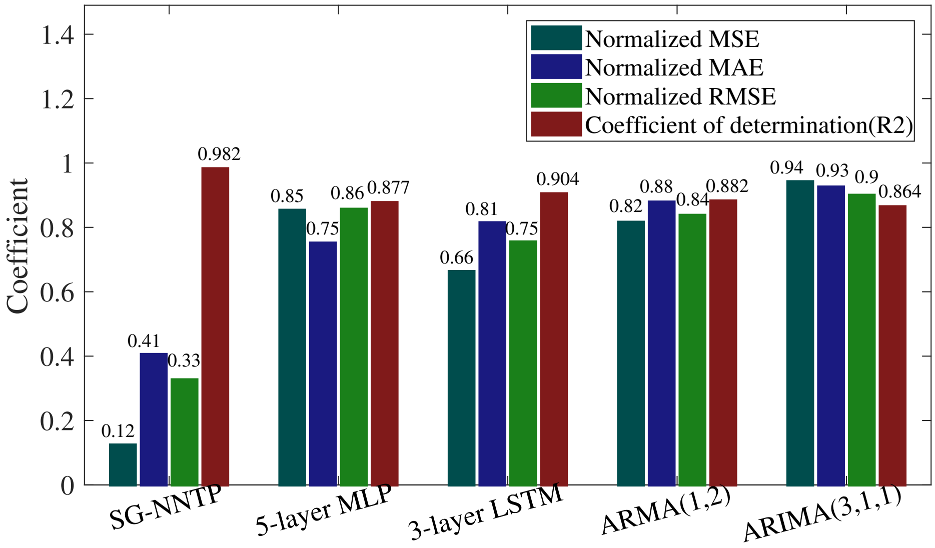


Fig. 4.20 The accuracy of the SG-NNTP model and the benchmark models in single-step prediction mode.

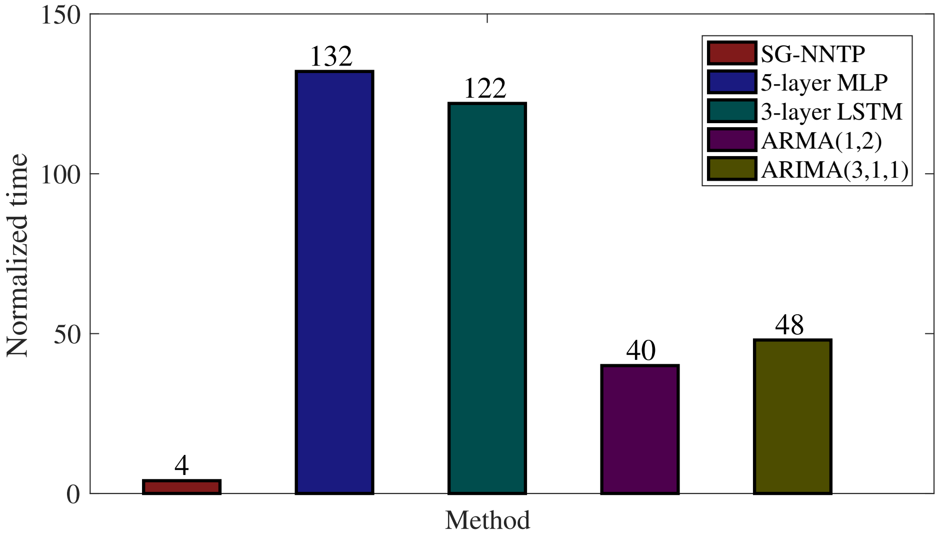


Fig. 4.21 The elapsed time of the SG-NNTP model and the benchmark models in single-step prediction mode.

is valuable for interdisciplinary study in the fields of communication and social science. The database can be used to identify the classification of newly given nonroutine traffic, discover the nonroutine events behind the traffic data, infer user behaviors, etc. In addition, analyzing the causes of formation and trends of nonroutine traffic can be very helpful for accurate and efficient network resource allocation.

What is more, the proposed NNTP method is essentially an efficient synthesis of multivariate data, which makes a recommendation for data collection and storage in the context of cellular network traffic. Specifically, additional information, such as regional events, number of users, etc, could be collected and stored with traffic data, which will be very beneficial for future research of nonroutine traffic. Similar events can potentially be further subdivided based on more detailed event information. In this way, more accurate NTP models with more reliable implicit relationships can be discovered. In fact, it is found that there exists some implicit relation between the parameter σ_{sg} and the number of attendances for all soccer games belonging to the Italian Serie A type. This relation is really helpful to initial parameter estimation and enable the model to achieve more accurate multi-step prediction results. However, due to the limitation of sample size, the implicit relationship has not been fully validated and therefore was not applied in current SG-NNTP model.

4.4 Conclusion

This chapter has raised the problem about nonroutine traffic, and pioneered a novel NNTP method to analyze and process nonroutine traffic. Subsequently, this chapter has constructed the SG-NNTP model for the additive nonroutine traffic caused by soccer games as a case study to validate the performance of the NNTP method. Experimental results show that the NNTP method outperforms the benchmark models in prediction accuracy, both in single-step and multi-step prediction mode. Also, the computational efficiency is greatly improved. In addition, the model constructed by the NNTP method is an analytical NTP model based on user behaviour analysis with outstanding interpretability. In our future research, we aim to integrate the NNTP model with information collection models to automate the gathering of imformation data of nonroutine events. The NNTP model will then process this data to achieve automatic prediction.

In general, both the UBB NTP method and the NNTP method are appropriate for the aggregate-level wireless network traffic data generated in larger regions rather than the cell-level wireless network traffic data, because the overall pattern of user behavior can be precisely extracted when the region contains sufficient users. Otherwise, the user behavior will exhibit apparent randomness. In this scenario, there is no choice but to employ ML-based model to perform cell-level NTP task. The next chapter focuses on the optimization of ML-based models in this scenario.

Chapter 5

Hyper-parameter Optimization for Cell-level Wireless Network Traffic Prediction with A Novel Meta-Learning Framework

Overview

In this chapter, we propose a novel cell-level wireless NTP framework, where an attention-based deep neural network (ADNN) is adopted as the prediction model, i.e., base-learner, for each cell-level mobile NTP task, namely base-task, and a meta-learner is employed to automatically generate the optimal hyper-parameters for a new base-learner according to the corresponding base-task's intrinsic characteristics or properties, i.e., meta-features. Based on the observation from real-world traffic records that base-tasks possessing similar meta-features tend to favour similar hyper-parameters for their base-learners, the meta-learner exploits the K-nearest neighbor (KNN) learning method to obtain a set of candidate hyper-parameter selection strategies for a new base-learner, which are then utilized by an advanced genetic algorithm with intelligent chromosome screening to finally acquire the best hyper-parameter selection strategy. Extensive experiments demonstrate that base-learners in the proposed framework have high potential prediction ability for cell-level mobile NTP tasks and the meta-learner can enormously elevate the base-learners' performance by providing them the optimal hyper-parameters.

5.1 Introduction

The previous two chapters have introduced two frameworks for daily and nonroutine wireless NTP tasks, respectively. However, the traffic patterns in cell-level wireless NTP tasks tends to exhibit a high degree of complexity due to the limited coverage area and the amount of users, as well as the mobility of users. Therefore, it poses a high demand on the NTP model's learning capacity. The deep learning-based NTP models possess the capacity to understand complex nonlinear relationships with the help of multiple activation functions, and thus match well with the cell-level wireless NTP tasks. However, the model's complexity is increased rapidly with its capacity. As a result, the hyper parameter selection of deep learning-based models is becoming a tricky problem.

Hyper-parameter selection involves numerous hyper-parameters including the learning rate, the number of neural layers, the number of neurons in each layer, etc., which have a significant influence on the models' after-training performance. Unfortunately, how to efficiently optimize the hyper-parameters for a cell-level mobile NTP model has not been well studied. As the state-of-the-art hyper-parameter optimization methods are general-purpose algorithms such as Genetic Algorithm (GA), PSO, and Bayesian Optimization. They lack specificity for NTP models. Consequently, when applied to NTP models, they face a significant trade-off between efficiency and performance, often requiring extensive computational time to achieve satisfactory results. Not to mention taking exhaustive searching method or a manual approach with expert experience. Furthermore, in the 5G or beyond mobile networks, the access points are ultra-densely deployed and there are tens of thousands of mobile cells to be considered in large-scale radio access networks [15] [16]. The large number of mobile cells significantly exacerbates these challenges.

With the objective of addressing the hyper-parameter optimization problem for NTP models, this work proposes a meta-learning framework for cell-level wireless NTP tasks to automatically optimize the hyper-parameters of a newly given cell-level NTP model based on the corresponding prediction task's intrinsic features (meta-features). The primary contributions of this chapter are summarized as follows:

- Using real-world cell-level mobile network traffic records generated in Milan, we statistically analyze the optimal hyper-parameter selection strategies of deep learning-based prediction models related to different cell-level traffic prediction tasks. By analyzing the information entropy of various hyper-parameter selections and the conditional entropy of hyper-parameter selection under various intrinsic characteristics or properties of prediction tasks, we conclude that these intrinsic characteristics or properties, i.e., meta-features, indeed have a significant influence on the distribution of optimal hyper-parameter selection strategies.

- A novel hyper-parameter optimization method is proposed for cell-level wireless NTP with a meta-learning framework. In the proposed framework, a cell-level wireless NTP task is seen as a base-task and an ADNN is introduced as the prediction model (base-learner) to address each base-task. We define finding the optimal hyper-parameters for the base-learner of each base-task as the meta-task. We present a meta-learner to handle the meta-task, which can automatically optimize the hyper-parameters of a base-task's base-learner according to the base-task's meta-features.

- The meta-learner exploits a KNN learning method to obtain a set of candidate hyper-parameter selection strategies for a new base-learner with the assistance of meta-knowledge accumulated from previous well-solved base-tasks. Then an advanced genetic algorithm with intelligent chromosome screening is presented to finally search the best hyper-parameter selection strategy by setting those candidate hyper-parameter selection strategies as partial of its first-generation chromosomes. Specifically, in order to elevate the modified genetic algorithm's computational efficiency, a sophisticated gated residual network (GRN) based deep neural network is presented to precisely evaluate the fitness value of each son chromosome.

- Compared with existing traffic prediction methods, the proposed framework has the following advantages. First, the proposed ADNN for cell-level wireless NTP can extract complex features and patterns from historical traffic data and thus has the potential to improve the model's prediction accuracy. Second, unlike the conventional deep learning-based NTP models whose hyper-parameters are provided randomly or through manual trial, the meta-learner in the proposed framework will generate the optimal hyper-parameters for each base-learner leading to the best after-training performance. We examine the performance of our framework through real-world cell-level traffic prediction tasks. Extensive experiments demonstrate that the meta-learner can elevate the base-learners' after-training prediction accuracy enormously by providing them with proper hyper-parameters.

The rest of this chapter is organized as follows. Existing research works on mobile NTP is reviewed in Section II. Section III introduces the real-world mobile network traffic records used in this chapter, followed by our statistically analysis about the prediction models' optimal hyper-parameter selection strategies. In Section IV, we describes the proposed meta-learning based cell-level traffic prediction framework in detail. Performance of our framework is evaluated in Section V. Finally, this chapter is concluded in Section VI.

5.2 Dataset and Preliminary Analyses

5.2.1 Cell-level Wireless Network Traffic Records

In this chapter, the wireless network traffic data in the "Telecom Italia Bia Data Challenge" from 01/11/2013 to 01/01/2014 in Milan [88] serves as the dataset. In the spatial dimension, Milan city is covered by 10000 grids, each of which possesses a size of $235m \times 235m$. Each traffic record in the dataset includes details regarding the occurrence time and volume of the network traffic, as well as the grid ID. As the coverage of a urban base station is close

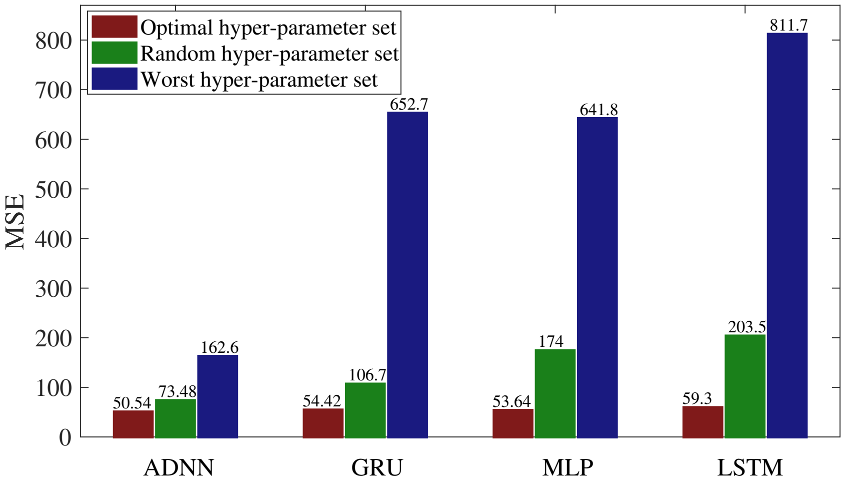


Fig. 5.1 The MSE performance of prediction models with different hyper-parameter selection strategies for mobile cell 1595.

to the size of the grid, we refer to each grid to a mobile cell [88]. The considered time span of the dataset is divided into 89,28 time intervals with the duration of ten minutes. Traffic load of the p-th cell ($p = 1$, ..., 10000) during the t-th time interval ($t = 1$, ..., $89,28$) can be acquired as $L_p[t]$ while the traffic load series of mobile cell p can be denoted as $\mathbf{L}_p = (L_p[1], L_p[2], ..., L_p[89,92])$. Specifically, in order to analyze the characteristics of traffic load series generated in different cells with a uniform scale, we normalize the elements in $\mathbf{L}_p$ into the range of [0,1] using the max-min normalization method as follows

$$\widetilde{L}_p[t] = \frac{L_p[t] - \min(\mathbf{L}_p)}{\max(\mathbf{L}_p) - \min(\mathbf{L}_p)}, \tag{5.1}$$

where $\max(\mathbf{L}_p)$ and $\min(\mathbf{L}_p)$ are the largest and smallest elements in $\mathbf{L}_p$, respectively. Accordingly, the normalized traffic load series of mobile cell p is denoted as $\widetilde{\mathbf{L}}_p$.

5.2.2 Preliminary Analyses of Hyper-parameter Selection

In this work, we regard forecasting a mobile cell's traffic load during a future time interval based on the loads in a number (step number) of previous time intervals as a cell-level

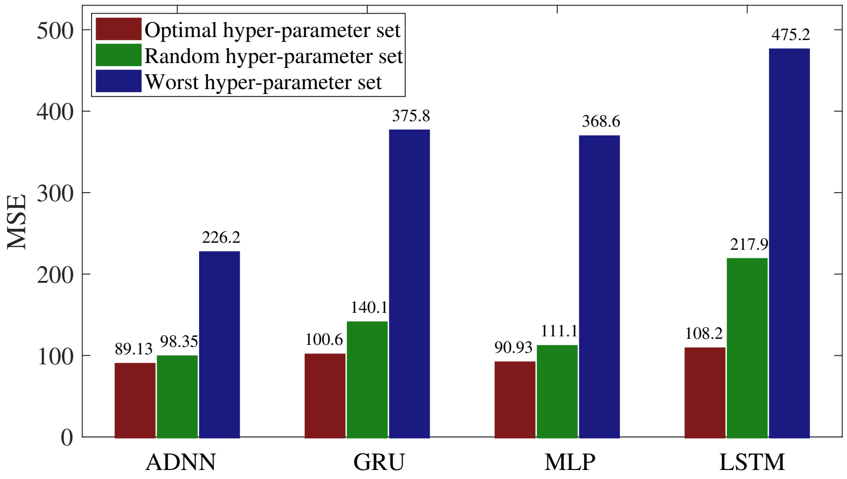


Fig. 5.2 The MSE performance of prediction models with different hyper-parameter selection strategies for mobile cell 2535.

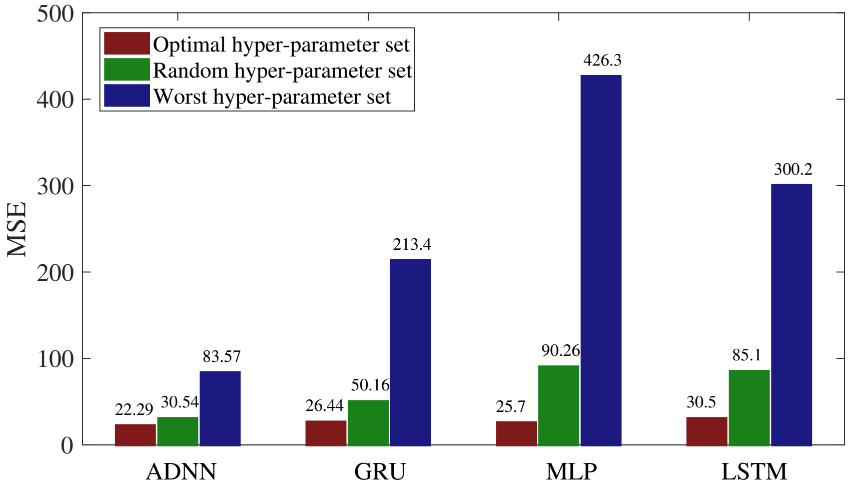


Fig. 5.3 The MSE performance of prediction models with different hyper-parameter selection strategies for mobile cell 3040.

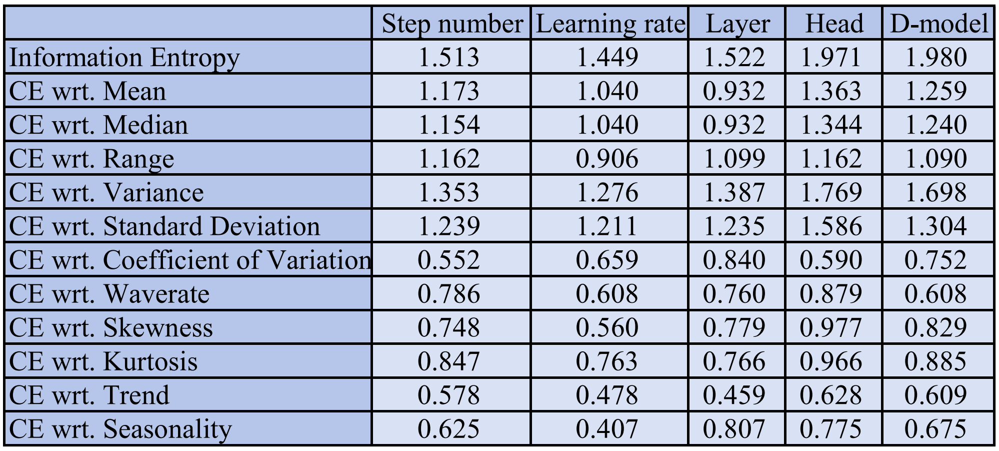
wireless NTP task. We apply a sliding window with size of the step number to split cell p's normalized traffic load series and generate samples for cell-level wireless NTP task p by labeling each split segment of traffic load series with the traffic load in the next time interval.

We first adopt four representative deep learning-based algorithms, i.e., the MLP network, the LSTM network, the gate recurrent unit (GRU) network, and the ADNN, to respectively construct the prediction model for each cell-level traffic prediction task and assess the impact of the hyper-parameter selection strategies on the prediction models' after-training performance. For different algorithms, the kinds of hyper-parameters and their selection ranges are listed in Table 5.1 and the hyper-parameters of ADNN is detailed in Section 5.3.2. Figs. 5.1, 5.2, and 5.3 show the MSE values of the well-trained prediction models related to three cell-level wireless NTP tasks (cells 1595, 2535, and 3040) over their testing samples when these models are with different algorithms and different hyper-parameter selection strategies. In Figs. 5.1, 5.2, and 5.3 the best or worst hyper-parameter selection strategy of each prediction model is acquired with the exhaustive searching method. From Figs. 5.1, 5.2, and 5.3, we make the following observations.

Observation 1: Adopting the best hyper-parameter selection strategies, the ADNN based prediction models seem to achieve a higher accuracy performance than prediction models with the other three learning algorithms over various prediction tasks. This may be explained by the fact that the ADNN can efficiently extract the complex temporal correlations among cell-level wireless network traffic loads generated in different time intervals.

Observation 2: Hyper-parameter selection strategies indeed have a great influence on the performance of prediction models. Furthermore, even with the same algorithm, the best hyper-parameter selection strategies vary a lot among prediction models related to different cell-level wireless NTP tasks.

We then test whether some intrinsic characteristics or properties of the cell-level wireless NTP tasks are correlated with the corresponding prediction models' best hyper-parameter selection strategies. We select the ADNN as those prediction models' learning algorithm and find the prediction models' best hyper-parameter selection strategies using the exhaustive searching method. Choosing candidate set of intrinsic characteristics or properties for a prediction task listed in Fig. 5.4, we calculate the conditional entropy of the best values of each kind of hyper-parameters over the prediction models with respect to each kind of intrinsic characteristics or properties [95], which is demonstrated in Fig. 5.4. As a comparison, the information entropies of the best values of different kinds of hyper-parameters over the prediction models are also provided. From Fig. 5.4, we have got the following observations.



	Step number	Learning rate	Layer	Head	D-model
Information Entropy	1.513	1.449	1.522	1.971	1.980
CE wrt. Mean	1.173	1.040	0.932	1.363	1.259
CE wrt. Median	1.154	1.040	0.932	1.344	1.240
CE wrt. Range	1.162	0.906	1.099	1.162	1.090
CE wrt. Variance	1.353	1.276	1.387	1.769	1.698
CE wrt. Standard Deviation	1.239	1.211	1.235	1.586	1.304
CE wrt. Coefficient of Variation	0.552	0.659	0.840	0.590	0.752
CE wrt. Waverate	0.786	0.608	0.760	0.879	0.608
CE wrt. Skewness	0.748	0.560	0.779	0.977	0.829
CE wrt. Kurtosis	0.847	0.763	0.766	0.966	0.885
CE wrt. Trend	0.578	0.478	0.459	0.628	0.609
CE wrt. Seasonality	0.625	0.407	0.807	0.775	0.675

Fig. 5.4 The information entropy and the conditional entropy (CE) of the best values of each kind of hyper-parameters.

Observation 3: Compared to the information entropy of the best values of each hyper-parameter, the conditional entropies with respect to different kinds of intrinsic characteristics or properties generally have smaller values, which demonstrates that these intrinsic characteristics or properties indeed possess obvious correlations with the prediction models' best hyper-parameter selection strategies and that cell-level wireless NTP tasks with similar intrinsic characteristics or properties tend to prefer similar hyper-parameters.

Observation 4: Different kinds of intrinsic characteristics or properties seem to have diverse importance on the distribution of the optimal hyper-parameter selection strategies of cell-level wireless NTP models.

5.3 The Proposed Cell-level Wireless NTP Framework

This section will give an overview of the proposed meta-learning based cell-level wireless NTP framework, followed by introduction of base-learners dealing with the cell-level wireless NTP tasks and the meta-learner, which can provide the best hyper-parameter selection strategies for different base-learners.

5.3.1 Framework Overview

Fig. 5.5 gives the diagram of our proposed cell-level wireless NTP framework. In the framework, each cell-level wireless NTP task is regarded as a base-task and ADNN based prediction models are presented as base-learners to hand various base-tasks. For a base-learner related to a specific base-task, there are several hyper-parameters, as listed in Table 5.1. According to the meta-learning theory [96], the hyper-parameter selection strategy determines a base-learner's hypothesis space comprised of all the hypothesis functions this base-learner can represent and how the training process will find a hypothesis function in this hypothesis space. A base-learner will have the potential to achieve high after-training prediction accuracy for a base-task when 1) its hypothesis space contains the hypothesis functions approaching the target function that perfectly fits the base-task's learning samples; and 2) a proper hypothesis function can be efficiently reached in the training process. As a result, the hyper-parameter selection strategy will seriously influence a base-learner's performance and the base-learners related to different base-tasks prefer different hyper-parameter selection strategies since these base-tasks possess diverse target functions and quite training samples.

Based on Fig. 5.4 as well as **Observations 3 and 4**, we choose I intrinsic characteristics or properties that most influence the optimal hyper-parameter selection strategies' distribution, i.e., introduce the lowest conditional entropies for various kinds of hyper-parameters, as each base-task's meta-features. In the proposed framework, we define the learning task of finding each base-learner's best hyper-parameter selection strategy according to the corresponding base-task's meta-features as the meta-task and present a meta-learner to solve it. We also construct a set of meta-samples to help the meta-learner handle the meta-task. Some notations used in the proposed framework are summarized as follows.

Notations in the proposed framework: $\mathscr{S}^{meta}$ represents the set of randomly selected meta-samples. What is more, $\mathscr{S}^{meta}$ is also used to represent the set of base-tasks to construct the meta-samples. s^{meta_p} represents the meta-sample in $\mathscr{S}^{meta}$, which is generated by base-task p corresponding to p-th mobile cell. $\mathscr{S}_{train}^{base_p}$ denotes the training set of base-samples for base-task p, while $\mathscr{S}_{valid}^{base_p}$ is base-task p's validation set of base-samples to evaluate the

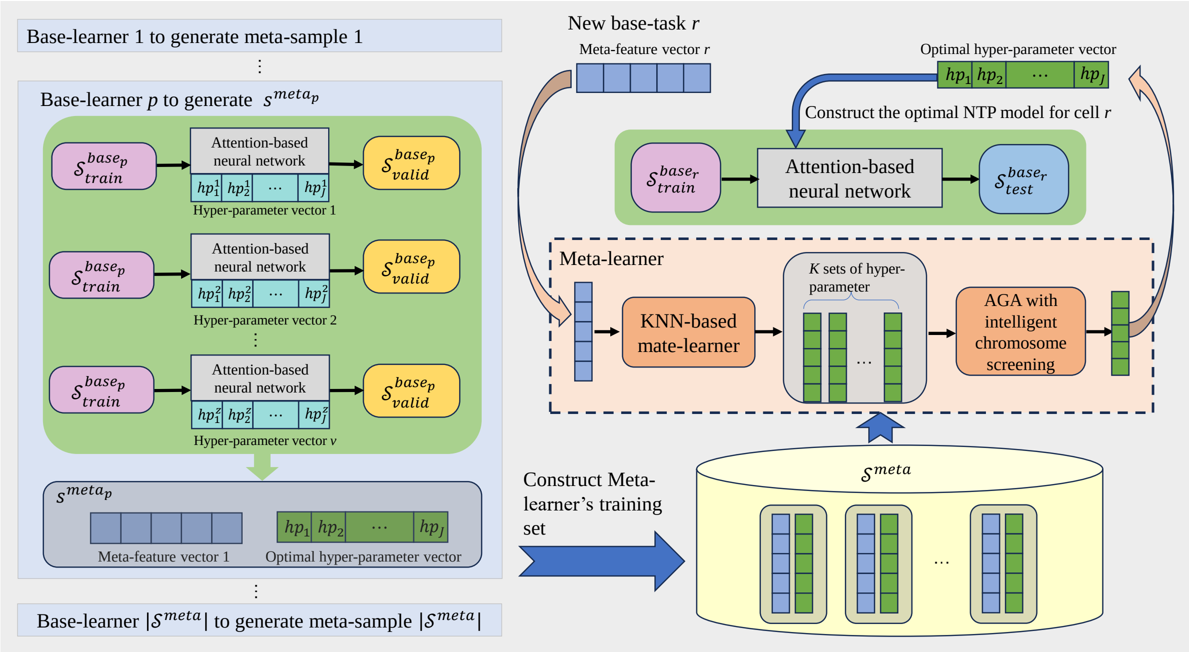

Base-learner 1 to generate meta-sample 1
Base-learner p to generate s^{meta_p}
$s^{base_p}_{train}$
Attention-based neural network
hp^1_1 hp^1_2 ⋯ hp^1_J
$s^{base_p}_{valid}$
Hyper-parameter vector 1
hp^2_1 hp^2_2 ⋯ hp^2_J
Hyper-parameter vector 2
hp^z_1 hp^z_2 ⋯ hp^z_J
Hyper-parameter vector v
s^{meta_p}
hp_1 hp_2 ⋯ hp_J
Meta-feature vector 1
Optimal hyper-parameter vector
Base-learner $|S^{meta}|$ to generate meta-sample $|S^{meta}|$
New base-task r
Meta-feature vector r
Optimal hyper-parameter vector
Construct the optimal NTP model for cell r
$s^{base_r}_{train}$
Attention-based neural network
$s^{base_r}_{test}$
Meta-learner
K sets of hyper-parameter
KNN-based mate-learner
AGA with intelligent chromosome screening
Construct Meta-learner's training set
s^{meta}

Fig. 5.5 The proposed cell-level wireless NTP framework with meta-learning.

fitness of different hyper-parameter selection strategies for base-learner p. For base-task $r \notin \mathscr{S}^{meta}$, the testing set of base-samples, $\mathscr{S}^{base_r}_{test}$, will also be constructed to examine its base-learner's after-training performance.

5.3.2 Base-learners

According to **Observation 1** that the attention mechanism can help prediction models extract temporal characteristics of cellular traffic patterns, we design the base-learner for each base-task as an ADNN as shown in Fig. 5.6. The ADNN is composed of an encoder and a decoder, which will be introduced in detail as follows.

Encoder

In a base-learner, the encoder contains L_e sequential encoder blocks with the same structure [97]. The input of the first encoder block is a $N_S \times D_{model}$ matrix, where the N_S row vectors represents a mobile cell's traffic loads generated in N_S continuous time intervals and the D_{model} is the dimension of the hidden layers of neural networks in encoder blocks and decoder blocks. In the Position Encoding layer, an additional dimension D_{model} is incorporated into the input matrix. Each encoder block possesses two sub-blocks. The first sub-block is the Multi-Head Attention layer. There are H_e Self-Attention structures in this layer and the key ($\mathscr{K}$), query ($\mathscr{Q}$), and value ($\mathscr{V}$) matrices in each Self-Attention structure can be calculated as

$$\begin{aligned}
\mathscr{K} &= \mathscr{E}_{in}^{(l_e)} \cdot \mathscr{W}_{\mathscr{K}}, \\
\mathscr{Q} &= \mathscr{E}_{in}^{(l_e)} \cdot \mathscr{W}_{\mathscr{Q}}, \\
\mathscr{V} &= \mathscr{E}_{in}^{(l_e)} \cdot \mathscr{W}_{\mathscr{V}},
\end{aligned} \tag{5.2}$$

where $\mathscr{E}_{in}^{(l_e)}$ is the input matrix of the l_e-th encoder block, $\mathscr{W}_{\mathscr{K}}$, $\mathscr{W}_{\mathscr{Q}}$, and $\mathscr{W}_{\mathscr{V}}$ are parameter matrices with the same dimension, which the model need to learn [97]. Fig. 5.7 shows the structure of each Self-Attention structure. The second sub-block is the Feed-Forward network consisting of two fully-connected layers of neural networks. The activation functions of the neurons in these two fully-connected layers take the ReLU function and the Linear function, respectively. Following each sub-block, the residual connection and batch normalization are appended sequentially. The output of each encoder block is a matrix with the size of $N_S \times D_{model}$ and will be seen as the input of the next encoder block. Specifically, we denote the output matrix of the last encoder block, $\mathscr{E}_{out}^{(L_e)}$, as the encoded matrix.

Decoder

As shown in Fig. 5.6, the decoder is constituted of L_e decoder blocks with the same structure [97]. The decoder block in our meta-learner contains three sub-blocks, i.e., Multi-Head Self-Attention layer, Multi-Head Encoder-Decoder Attention layer, and a Feed-Forward network, as shown in Fig. 5.6. The Multi-Head Self-Attention layer takes the same input matrix of the encoder as its input. It has H_e Self-Attention structures, each of which processes the input matrix using the same method as any Self-Attention Structure in an encoder block. We denote the output matrix of the Multi-Head Self-Attention layer as $\mathscr{D}$. $\mathscr{D}$ has the same dimension with the input matrix of the encoder. The Multi-Head Encoder-Decoder Attention layer takes both $\mathscr{E}_{out}^{(L_e)}$ and $\mathscr{D}$ as the inputs. It has a similar structure with the former Multi-Head

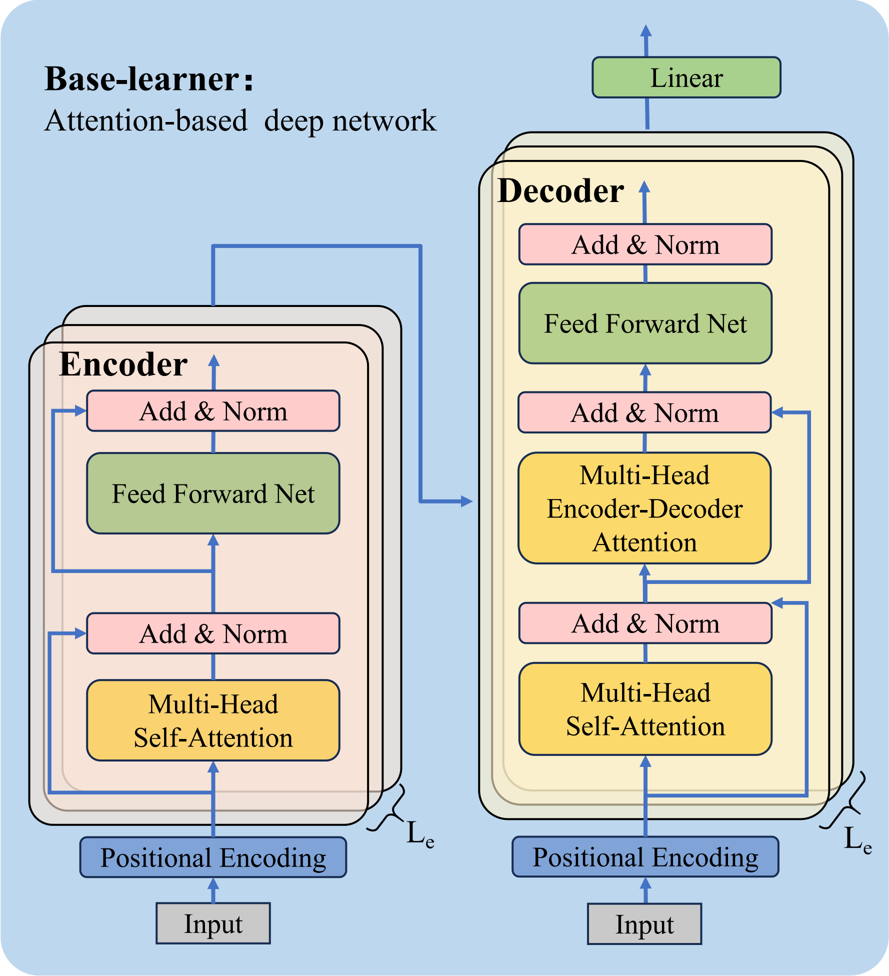


Fig. 5.6 The structure of the base-learner.

Attention layer. However, query ($\mathscr{Q}$) matrix is generated by $\mathscr{D}$, while the key ($\mathscr{K}$) and value ($\mathscr{V}$) matrices are generated by $\mathscr{E}_{out}^{(L_e)}$ with the following equations:

$$\begin{aligned} \mathscr{K} &= \mathscr{E}_{out}^{(L_e)} \cdot \mathscr{W}_{\mathscr{K}}, \\ \mathscr{Q} &= \mathscr{D} \cdot \mathscr{W}_{\mathscr{Q}}, \\ \mathscr{V} &= \mathscr{E}_{out}^{(L_e)} \cdot \mathscr{W}_{\mathscr{V}}. \end{aligned} \tag{5.3}$$

Finally, the Feed-Forward network takes the same structure as that in any encoder block. Also, the residual connection and batch normalization are appended following each sub-block in the decoder block.

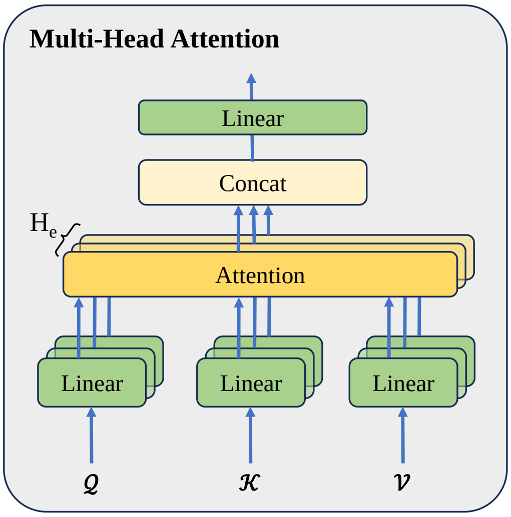


Fig. 5.7 The schematic diagram of the multi-head attention mechanism.

The fully-connected layer then transforms the output matrix of the decoder block into a vector and the neurons in this layer take the ReLU function as their activation functions. The output layer of the decoder consists of only one neuron representing the prediction result for the mobile cell's traffic load in next time interval.

For each base-learner, we define L_e, N_S, D_{model}, H_e, and the learning rate in the training process, c, as its hyper-parameters.

5.3.3 The Proposed Meta-learner

Based on **Observation 3** that base-tasks with similar meta-features prefer similar hyper-parameters, we design a two-stage meta-learner, which finds a set of high-quality candidate hyper-parameter selection strategies for the base-learner of a newly considered base-task with the KNN learning method in the first stage and then achieves the base-learner's optimal hyper-parameter selection strategy using an advanced genetic algorithm with intelligent chromosome screening module in the second stage.

KNN Learning Method

In order to leverage the KNN learning algorithm, a set of meta-samples, $\mathscr{S}^{meta}$, is constructed with $|S^{meta}|$ randomly selected base-tasks. $|S^{meta}|$ is the cardinality of set $\mathscr{S}^{meta}$. For each meta-sample s^{meta_p} in $\mathscr{S}^{meta}$, we build it by labeling base-task p's meta-features with the best hyper-parameter selection strategy of its base-learner, which is acquired with the exhaustive searching method.

Since different meta-features have different levels of importance in the hyper-parameter selection, it is inappropriate to directly use the Euclidean distance between feature vectors to represent the distance from base-task p to base-task r. Hence, we introduce an MLP network to perform linear and nonlinear processing of the meta-feature vectors of s^{meta_r} and s^{meta_p} to obtain the distance D_{p-to-r} between s^{meta_r} and s^{meta_p}. While their real distance RD_{p-to-r} is represented as the performance achieved when the s^{meta_r} adopts the meta-label vector of s^{meta_p}. Finally, the MLP network is trained by minimizing the error between D_{p-to-r} and RD_{p-to-r}.

For a newly considered base-task r, the KNN learning method finds K meta-samples from $\mathscr{S}^{meta}$, whose meta-feature vectors are with the shortest distances with that of base-task r. Since the K hyper-parameter selection strategies related to the K picked meta-samples are expected to provide good after-training performance for base-learner r, they will be regarded as candidate hyper-parameter selection strategies for base-learner r, which will also be set as first-generation chromosomes in the following advanced genetic algorithm.

Advanced Genetic Algorithm with Intelligent Deep Learning Assisted Chromosome Screening

Obviously, the hyper-parameter selection space for each base-learner is quite huge and it is almost impossible to establish a close-form mapping between the base-learner's after-training performance and the hyper-parameter selection strategy. Inspired by the fact that the genetic algorithm can efficiently search the solution spaces of complex optimization problems and requires no information about the forms of objective functions [98], we propose an advanced

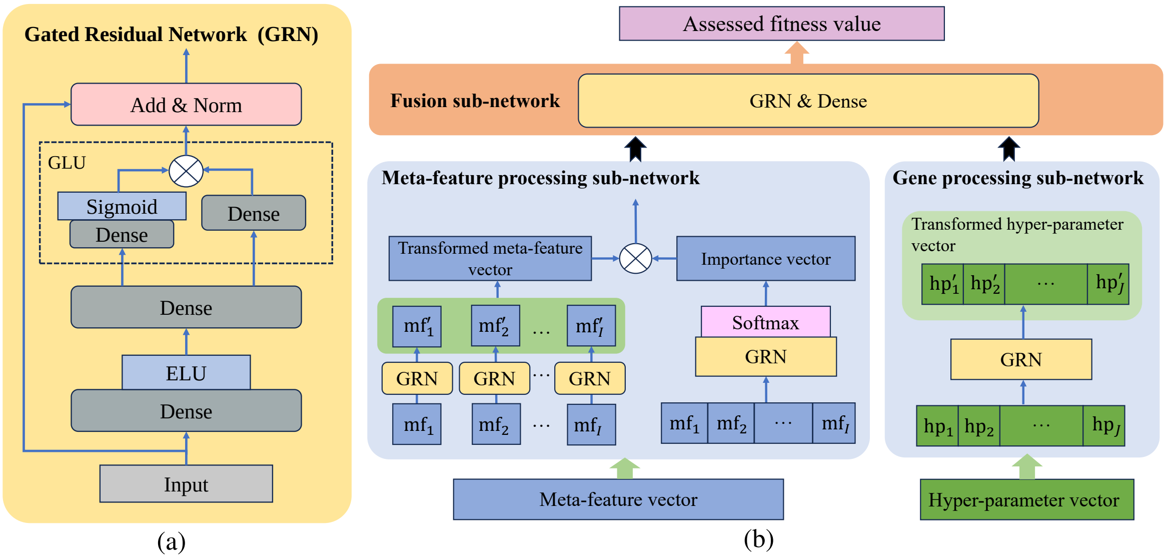


Fig. 5.8 a) The structure of Gated Residual Network; b) The framework of GRN module.

genetic algorithm to finally find the best hyper-parameter selection strategy for base-learner of the newly considered base-task $r \notin \mathscr{S}^{meta}$ (base-learner r).

Specifically, the advanced genetic algorithm regards each kind of hyper-parameters as a fragment of one chromosome (gene) and regards each possible hyper-parameter selection strategy of a base-learner as one chromosome. A chromosome's fitness value is defined as the reciprocal of base-learner r's generalization error over $\mathscr{S}^{base_r}_{valid}$ when base-learner r adopts the hyper-parameter selection strategy provided by this chromosome and is well-trained with $\mathscr{S}^{base_r}_{train}$. Besides the K hyper-parameter selection strategies generated by the KNN learning method, the advanced genetic algorithm also generates $M-K$ chromosomes, in each of which the value of any gene is randomly assigned over the corresponding hyper-parameter's selection range with a uniform distribution, as its first-generation chromosomes.

The advanced genetic algorithm reaches its final solution through N generations of chromosomes and there will be M chromosomes surviving in each generation. We denote the M remaining chromosomes in the n-th ($n = 1, \dots N-1$) generation as $\zeta_1^{(n)}$, ..., $\zeta_M^{(n)}$, whose fitness values are $f_1^{(n)}$, ..., $f_M^{(n)}$, respectively. Based on these M chromosomes, W ($W >> M$) son chromosomes are built. In these W son chromosomes, $p_{rem} \cdot W$ ones are obtained by directly duplicating $p_{rem} \cdot W$ parent chromosomes from $\zeta_1^{(n)}$, ..., $\zeta_M^{(n)}$ with the highest fitness values. The other $(1-p_{rem})\cdot W$ son chromosomes are hybrid ones, genes of each of which are inherited and crossed from two parent chromosomes selected randomly from $\zeta_1^{(n)}$, ..., $\zeta_M^{(n)}$. In order to prevent the algorithm from falling into a local optimum, any gene of every hybrid son chromosome possesses a probability of p_{mut} to mutate into a random value in its selection range with a uniform distribution. Specifically, all the parent chromosomes have the same probability to be chosen as one of a hybrid son chromosome's parent chromosomes.

To conquer the challenge that calculating the fitness values of W son chromosomes for each chromosome generation (training base-learner r W times with different hyper-parameter selection strategies) is quite computationally complex, an intelligent deep learning assisted chromosome screening scheme is proposed for the advanced genetic algorithm to find the M $(n+1)$-th generation surviving chromosomes. The advanced genetic algorithm utilizes a sophisticated GRN based deep neural network [99] as shown in Fig. 5.8 to evaluate fitness values of the W son chromosomes and selects $\tau\cdot M$ son chromosomes with the highest evaluated fitness values, where τ is a constant larger than 1. After that, fitness values of only those $\tau\cdot M$ son chromosomes are calculated and the M ones with the largest actual fitness values survive as the next-generation chromosomes. Please note that since $\tau\cdot M\ll W$, the novel intelligent deep learning assisted chromosome screening scheme will improve the advanced genetic algorithm's computational efficiency tremendously.

As shown in Fig. 5.8 (b), the proposed GRN based deep neural network takes base-task r's meta-features and hyper-parameter selection strategy related to a son chromosome as its inputs while outputs the evaluated fitness value for this son chromosome. With motivation of giving the deep neural network flexibility to apply linear processing and non-linear processing of its inputs only where needed, the GRN structure is presented in Fig. 5.8 (a) as a building block of the deep neural network. A GRN block ω takes in an input vector $\mathbf{i}$ and yields

$$GRN_{\omega}(\mathbf{i})=LayerNorm(\mathbf{i}+GLU_{\omega}(\eta_1)), \tag{5.4}$$

$$\eta_1=\mathbf{O}_{1,\omega}\eta_2+\mathbf{b}_{1,\omega}, \tag{5.5}$$

$$\eta_2 = ELU\left(\mathbf{O}_{2,\omega}\mathbf{i} + \mathbf{b}_{2,\omega}\right), \tag{5.6}$$

where ELU is the exponential linear unit activation function [99]; LayerNorm is a standard normalization layer; η_1 and η_2 are intermediate layer outputs. Taking η_1 as the input, the gated linear units (GLU) module generates

$$GLU_{\omega}\left(\eta_1\right) = \sigma\left(\mathbf{O}_{3,\omega}\eta_1 + \mathbf{b}_{3,\omega}\right) \odot \left(\mathbf{O}_{4,\omega}\eta_1 + \mathbf{b}_{4,\omega}\right), \tag{5.7}$$

where $\sigma(\cdot)$ is the sigmoid activation function, $\odot$ is the element-wise Hadamard product. In equations (5-7), $\mathbf{O}_{(\cdot)}$ and $\mathbf{b}_{(\cdot)}$ are the neuron connection weight matrix and neuron bias vector, respectively. The GLU module allows a GRN block to control the extent to which the non-linear processing of the input vector **i** contributes to the output vector, e.g., the GLU outputs could be close to 0 in order to suppress the nonlinear contribution.

The GRN based deep neural network demonstrated in Fig. 5.8 (b) contains three sub-networks: the meta-feature processing sub-network, the gene processing sub-network, and the fusion sub-network. The meta-feature processing sub-network takes base-task *r*'s meta-features as its input vector and applies linear and non-linear processing for the inputs via GRN blocks, generating the transformed meta-features. Due to the facts that different kinds of meta-features have diverse ranging scales and diverse importance on base-learner *r*'s performance, we also introduce an automatic importance evaluation mechanism for the meta-features in this sub-network. Specifically, taking base-task *r*'s meta-features as the inputs, an importance vector is generated through a GRN block and a Softmax layer. The meta-feature processing sub-network then takes the Hadamard product between the transformed meta-features and the importance vector as its outputs. The gene processing sub-network takes the genes determined by a son chromosome as its inputs and yields the transformed gene values via multiple GRN blocks. Finally, the fusion sub-network conducts linear as well as non-linear processing for the outputs of the former two sub-networks via a GRN block, and yields the evaluated fitness values of a son chromosome for base-learner r through a full-connection neural network.

We train the GRN based deep neural network with learning samples generated from base-tasks in $\mathscr{S}^{meta}$. For any base-task $p \in \mathscr{S}^{meta}$, since the fitness values of all the possible hyper-parameter selection strategies have been tested for base-learner p, numerous learning samples can be built related to this base-task for the GRN based deep neural network by labeling each hyper-parameter selection strategy as well as base-task p's meta-features with the corresponding fitness value.

Finally, we present pseudo codes in **Algorithm 2** to illustrate the meta-learner's operating process in detail.

5.4 Numerical Results of the Proposed Framework

This section will first introduce our experimental settings and the performance metrics we use. Then, we numerically prove effectiveness of the KNN learning method adopted in the proposed meta-learner and how the key parameters will influence the advanced genetic algorithm's performance. After that, prediction accuracy and computational complexity of the proposed framework is compared with several benchmark methods. At last, robustness of **Algorithm 2** is analyzed.

5.4.1 Experimental Settings

In the adopted dataset, traffic load records of some mobile cells during a number of time intervals are missing due to collection failure or storage error. When we have not got the actual traffic load of a mobile cell during a certain time interval, we will fill it based on a widely used method [100] where the missing value is completed with the average traffic load of the target cell's eight surrounding cells during the same time interval.

We compare the performance of our framework with several representative cell-level wireless NTP methods. These methods include the SVR, the Gaussian processing, and conventional deep learning-based methods like the MLP network, the LSTM network,

Algorithm 2:

```
Input: $\mathscr{S}^{meta}, p_{rem}, p_{mut}$, base-task $r$
 1: Get meta-feature vector of base-task $r$;
 2: Obtain $K$ neighbouring meta-samples from $\mathscr{S}^{meta}$ whose meta feature vectors have the
    smallest distances with the meta-feature vector of base-task $r$ with the KNN learning
    algorithm;
 3: Generate $K$ first-generation chromosomes ($K$ hyper-parameter selection strategies for
    base-learner $r$) with the labels of the $K$ selected meta samples;
 4: Generate $M - K$ first-generation chromosomes randomly;
 5: Construct the set of $M$ first-generation chromosomes based on the outputs of steps 3 and
    4;
 6: Calculate the fitness value of each first-generation chromosome;
 7: for $n = 1 : N$ do
 8:     Select $p_{rem} \cdot W$ chromosomes in the $n$-th generation ones with the largest fitness
        values as the son chromosomes;
 9:     for $w = 1 : (W - p_{rem} \cdot W)$ do
10:         Select two chromosomes in the $n$-th generation ones randomly with the uniform
            selecting probability;
11:         Generate a son chromosome by inheriting and crossing the genes of the two above
            selected chromosomes;
12:         Mutate each gene of the son chromosome with a probability of $p_{mut}$ into a random
            value in the gene's selection range;
13:     end for
14:     Construct the set of $W$ son chromosomes for the $n$-th generation ones based on the
        outputs of steps 10-12;
15:     Evaluate the fitness values of the $W$ son chromosomes with the proposed GRN deep
        neural network;
16:     Select $\tau \cdot M$ son chromosomes with the largest evaluated fitness values;
17:     Calculate the actual fitness values of $\tau \cdot M$ son chromosomes and obtain $M$ son
        chromosomes with the largest actual fitness values as the survived ones $(n+1)$-th
        generation chromosome;
18: end for
Output: The chromosome having the largest fitness value within $N$ generations of
    chromosomes
```

Table 5.1 The hyper-parameters' selection range.

	Step number (N_S)	Learning rate (c)	Layer (L_e)
ADNN	(6, 12, 18)	(0.01, 0.001, 0.0001)	(1, 2, 3)
	Step number	Learning rate	Layer
GRU	(6, 12, 18)	(0.01, 0.001, 0.0001)	(2, 3, 4)
LSTM	(6, 12, 18)	(0.01, 0.001, 0.0001)	(2, 3, 4)
MLP	(6, 12, 18)	(0.01, 0.001, 0.0001)	(2, 3, 4)

	Head (H_e)	D_{model}
ADNN	(2, 4, 6, 8)	(8, 16, 32, 64, 128, 256, 512)
	Neure	
GRU	(256, 512, 768)	
LSTM	(32, 64, 128, 256)	
MLP	(128, 256, 512)	

the GRU network, and the ADNN as shown in Fig. 5.6. In order to test how the meta-learning technology in the proposed framework can improve the base-learners' after-training performance by providing the proper hyper-parameters, the conventional deep learning-based methods are with randomly selected hyper-parameters for the related prediction models in handling each base-task, where the hyper-parameters' selection ranges are listed in Table I. For the ADNN based base-learners, we also test their performance when their hyper-parameters are provided by the meta-learner presented in **Algorithm 2**, the GA with no deep learning assisted chromosome screening, the advanced genetic algorithm with deep learning assisted chromosome screening but randomly selected fist-generation chromosomes (AGA), and the genetic algorithm with fist-generation chromosomes selected by the KNN learning method but with no deep learning assisted chromosome screening (GA+KNN), as well as when their hyper-parameters are optimized by the exhaustive searching method (ES).

We randomly select 160 mobile cells in our dataset to construct the set of meta-samples for the proposed framework, $\mathscr{S}^{meta}$, and regard the remaining mobile cells as the testing base-tasks. For mobile cell $p \in \mathscr{S}^{meta}$, the training set and validation set of its base-learner are constructed by base-samples generated in range of 01/11/2013-30/11/2013. For mobile cell $r \notin \mathscr{S}^{meta}$, the training set and validation set are constructed by base-samples generated in range of 01/11/2013-30/11/2013, while the testing set is composed of base-samples generated in range of 01/12/2013-03/12/2013. With the aim of fair comparison, training samples for learning models of the considered conventional deep learning-based prediction methods related to each testing base-task are generated in range of 01/11/2013-30/11/2013.

Base-learners in the proposed framework as well as the deep learning-based prediction models are optimized with a stochastic gradient based optimization technique, AdamW [101], which is widely adopted in ML domain. The MSE is chosen as the loss function in the learning models' training process. To evaluate the prediction methods' accuracy performance, two metrics, i.e., MSE and R2, are adopted in our experiments. Specifically, R2 will measure the fitting degree between the prediction and ground true values.

5.4.2 Effectiveness of the KNN Learning Method

This subsection will experimentally verify the effectiveness of the KNN learning method adopted by the meta-learner in deriving high-quality fist-generation chromosomes.

For the testing base-tasks, Fig. 5.9 presents the average after-training MSE and R2 performance achieved by their base-learners over the validation sets when each base-learner adopts the best hyper-parameter selection strategy from the candidate strategies provided by the corresponding base-task's neighbor meta-samples in $\mathscr{S}^{meta}$ versus the neighbor number K in the KNN learning algorithm under different scales of $\mathscr{S}^{meta}$. Specifically, a smaller MSE or a larger R2 reflects that the KNN learning algorithm in the meta-learner can provide better candidate hyper-parameter selection strategies for the base-learners of testing base-tasks. We can observe from Fig. 5.9 that under a given $\mathscr{S}^{meta}$ size, performance of the KNN learning algorithm upgrades transparently as K augments. This is because when we consider more neighbor meta-samples for a testing base-task, the KNN learning will have a larger probability to obtain the competent hyper-parameter selection strategy from these meta-samples' labels for the corresponding base-learner to achieve higher after-training prediction accuracy. An interesting phenomenon in Fig. 5.9 is that as K gets large from 3 to 10, the KNN learning algorithm's performance with a certain scale of $\mathscr{S}^{meta}$ will improves swiftly at first and then become stable. Specifically, when K exceeds 8, further augment of K only introduces little performance rising for the KNN learning algorithm.

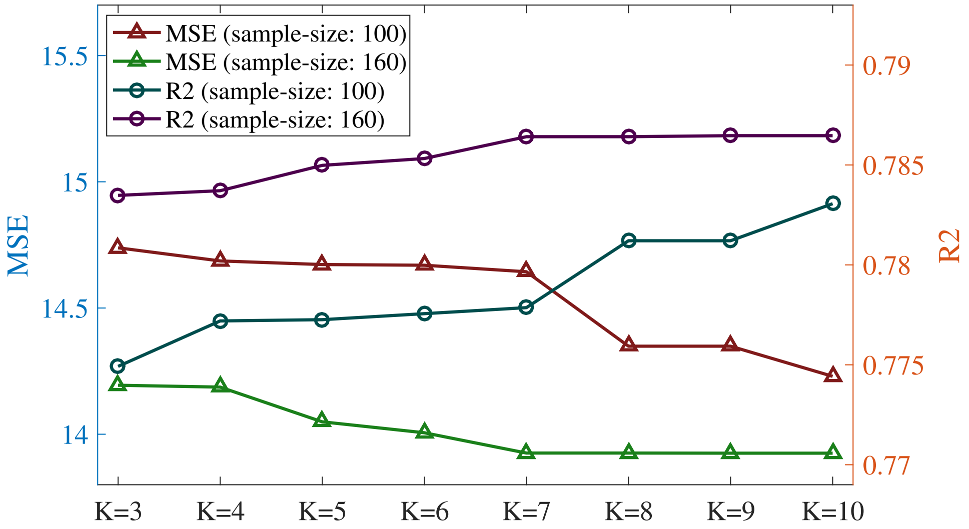


Fig. 5.9 The average after-training MSE and R2 performance achieved by base-learners when each base-learner adopts the conditionally optimal hyper-parameter selection strategy provided by the KNN learning algorithm versus the neighbor number K under different scales of $\mathscr{S}^{meta}$.

Fig. 5.10 shows performance of the KNN learning algorithm versus the scale of $\mathscr{S}^{meta}$ with different values of K. Fig. 5.10 demonstrates that under a certain selection of K, the KNN learning algorithm will always perform better when $\mathscr{S}^{meta}$ contains more meta-samples. This can be explained as when more base-tasks are solved and more meta-knowledge is accumulated, it will be easier for the KNN learning algorithm to find meta-samples possessing similar meta-features with any given testing base-task. As a result, these meta-samples tend to provide more satisfactory hyper-parameter selection strategies for the related testing base-learner according to **Observation 4** that base-tasks with similar meta-features are likely to prefer similar hyper-parameters. Moreover, as illustrated in Fig. 5.10, we can observe that the KNN learning algorithm's effectiveness will gradually flatten out after $\mathscr{S}^{meta}$'s scale is larger than 140, which imply that it seems not necessary to acquire excessively much meta-knowledge to guarantee the performance of the KNN learning algorithm.

In the rest of our experiments, we will set K and $\mathscr{S}^{meta}$'s scale as 8 and 160, respectively, to balance the framework performance and the calculation complexity.

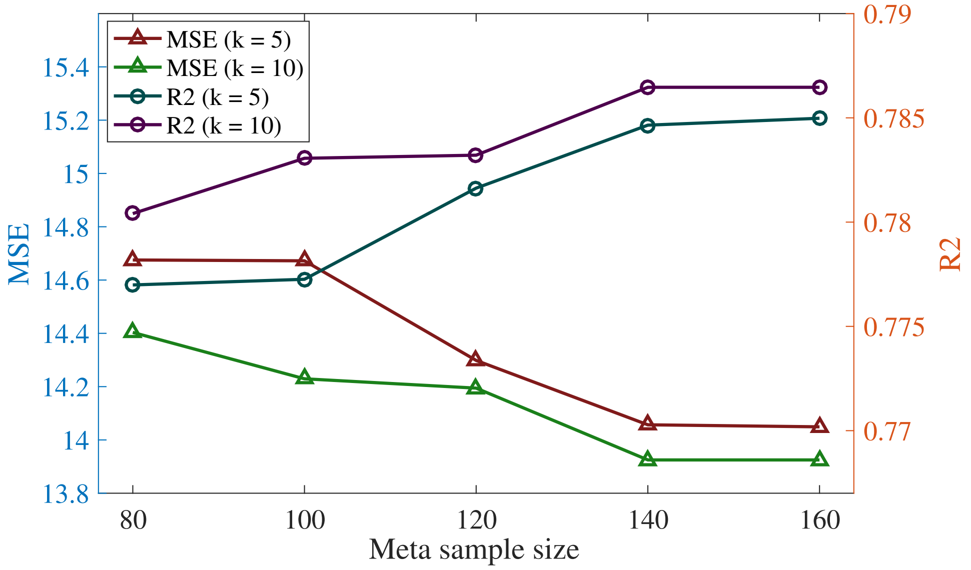


Fig. 5.10 The average after-training MSE and R2 performance achieved by base-learners when each base-learner adopts the conditionally optimal hyper-parameter selection strategy provided by the KNN learning algorithm versus the scale of $\mathscr{S}^{meta}$ with different values of K.

5.4.3 Influence of Key Parameters in the Advanced Genetic Algorithm

Proportion of parent chromosomes remaining in the W candidate son chromosomes, p_{rem}, and mutation probability of genes, p_{mut}, are two key parameters of the proposed advanced genetic algorithm in the meta-learner. This subsection will test how these two parameters influence the advanced genetic algorithm's performance when part of its first-generation chromosomes are generated with assistance of the KNN learning method.

For two randomly selected testing base-tasks (mobile cells 1635 and 4004), Figs. 5.11 and 5.12 show the corresponding base-learners' after-training accuracy performance with the currently optimal hyper-parameter selection strategies provided by the advanced genetic algorithm versus the algorithm's processing time when p_{rem} and p_{mut} have some certain values. Specifically, results of each curve in Fig. 5.11 or Fig. 5.12 are averaged over 10 random tests of the algorithm. From these two figures, we can observe that the algorithm will have faster converging speed but lower after-convergence performance when p_{rem} possesses a larger value or p_{mut} possesses a smaller value. Reasons behind this phenomenon

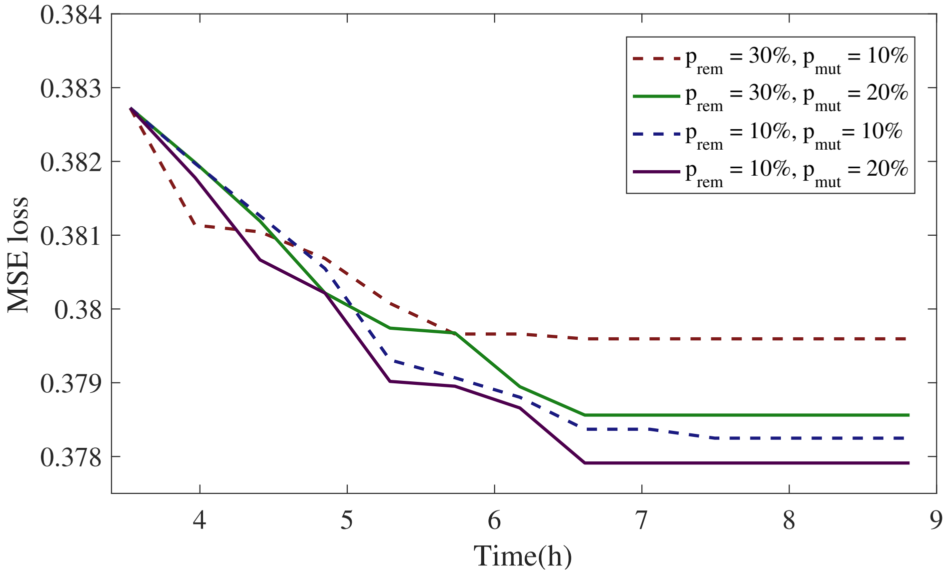


Fig. 5.11 The base-learner's after-training MSE performance with the currently optimal hyper-parameter selection strategy output by **Algorithm 2** for mobile cell 1635 versus **Algorithm 2**'s processing time under different value combinations of p_{rem} and p_{mut}.

can be explained as follows. When p_{rem} is large and p_{mut} is small, more high-quality parent chromosomes and advantageous genes will be preserved in the next-generation chromosomes. As a result, the algorithm can rapidly find out a proper solution of the hyper-parameter optimization problem for each testing base-task. Moreover, as more parent chromosomes remain in the set of survived next-generation chromosomes, the algorithm can avoid calculating their fitness values in this iteration, which costs enormous processing time to train the base-learner with various hyper-parameter selection strategies, since these values have been obtained previously. However, due to the fact that the algorithm tends to utilize existing chromosomes and genes with a larger p_{rem} or a smaller p_{mut}, it may be stuck at a local optimum and not output a satisfactory solution. On the other hand, Figs. 5.11 and 5.12 show that the advanced genetic algorithm obtains better solutions with lower converging speed when p_{rem} is small and p_{mut} is large. This is because exploring the solution space more extensively in each iteration increases the likelihood of finding global optimums, albeit at the cost of longer searching time.

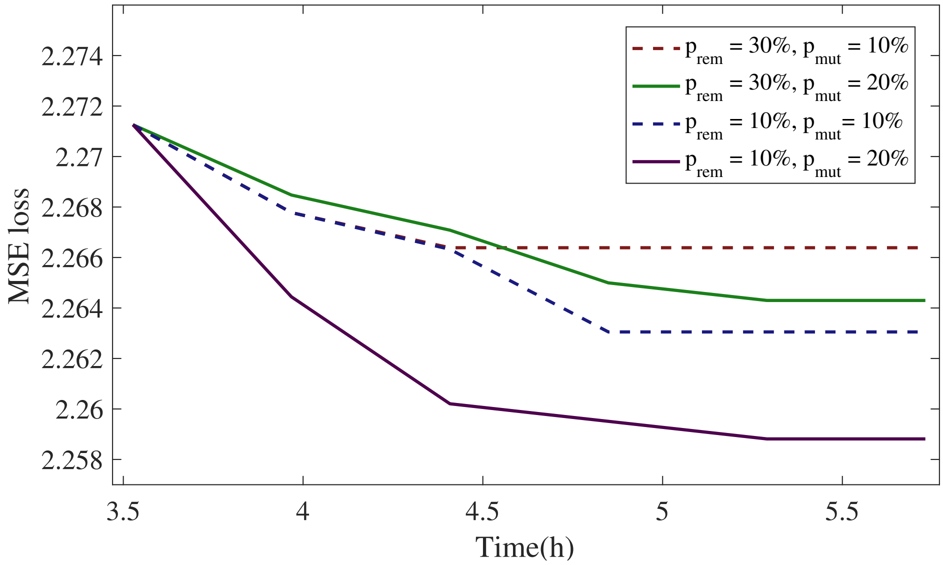


Fig. 5.12 The base-learner's after-training MSE performance with the currently optimal hyper-parameter selection strategy output by **Algorithm 2** for mobile cell 4004 versus **Algorithm 2**'s processing time under different value combinations of p_{rem} and p_{mut}.

Even with different values of p_{rem} and p_{mut}, we can observe from Figs. 5.11 and 5.12 that the proposed advanced genetic algorithm will further optimize the hyper-parameter selection strategies generated by the KNN learning method and converge with acceptable processing time. Specifically, we select a relatively small p_{rem} and a relatively large p_{mut} ($p_{rem} = 10\%$, $p_{mut} = 20\%$) to elevate the algorithm's after-convergence performance.

5.4.4 Prediction Accuracy of the Proposed Framework and Benchmark Methods

We compare the after-training accuracy performance of the proposed framework and the benchmark methods. The MSE and R2 values of the considering prediction methods in handling four randomly selected testing base-tasks as well as averaged over all the testing base-tasks are demonstrated in Table 5.2.

As shown in Table 5.2, the shallow learning-based methods, SVR and Gaussian processing, exhibit relatively low prediction accuracy, though they have the ability of extracting the nonlinearities in time series. This can be explained as cell-level wireless network traffic patterns contain complex auto-correlations in the temporal domain, which might exceed the learning capacity of the shallow learning-based methods. As numbers of coefficients can be tuned in SVR and Gaussian processing are quite limited, it is hard for them to present the cell-level wireless network traffic characteristics. Consequently, SVR and Gaussian processing will face the under-fitting problem and not obtain satisfactory accuracy performance. The complex learning models and the huge numbers of parameters which need to be adjusted in the conventional deep learning-based methods, i.e., MLP network, LSTM network, GRU network, and ADNN, make them possess huge potential to explore deep dependencies among cell-level wireless network traffic loads in different time intervals. As a result, these conventional deep learning-based methods achieve lower average MSE and higher average R2 over the testing base-tasks than SVR and Gaussian processing. We can also observe from Table 5.2 that the conventional deep learning-based methods present quite unstable after-training accuracy performance in dealing with different base-tasks, e.g., the MLP network introduces large prediction error for base-task 6035 while the ADNN introduces large prediction error for base-task 9106. This is because as shown in **Observation 2**, hyper-parameter selection strategies highly influence the deep learning models' performance and inappropriate hyper-parameter selection strategies may hinder their prediction accuracy seriously.

Our proposed framework achieves very high prediction accuracy and performs stably in handling the testing base-tasks. Reasons behind this phenomenon can be explained in two folds. First, base-learner of each base-task in the proposed framework is designed as an ADNN, which has the potential of learning and extracting the complex temporal correlations and nonlinearities hidden in cell-level wireless network traffic load series. Second, unlike the conventional deep learning-based prediction models, whose hyper-parameters are fixed or randomly selected for various base-tasks, meta-learner in the proposed framework will provide the appropriate hyper-parameter selection strategy for base-learner of each testing base-task according to the base-task' meta-features, making the base-learner's hypothesis space contain proper hypothesis functions similar to the considered base-task' target function

Table 5.2 The performance of the proposed framework and the benchmark methods

Benchmark methods without hyper-parameter optimization										
	cell 6035		cell 6036		cell 8417		cell 9106		Mean	
	MSE	R2	MSE	R2	MSE	R2	MSE	R2	MSE	R2
SVR	5.12	41.58%	61.14	22.02%	122	46.22%	58.62	20.64%	36.21	44.96%
GP	6.45	26.34%	50.02	36.21%	148.1	34.71%	31.48	57.38%	41.27	40.37%
GRU	3.52	59.86%	66.05	15.76%	81.31	64.16%	15.65	78.81%	24.71	62.6%
LSTM	5.33	39.13%	49.24	37.19%	96.58	57.43%	22.66	69.32%	28.68	52.39%
MLP	5.8	33.82%	67.83	13.48%	78.13	65.56%	19.04	74.22%	26.2	55.71%
ADNN	4.28	51.14%	42.95	45.22%	88.25	61.1%	37.36	49.43%	23.98	64.79%
Traditional Hyper-parameter Optimization method										
GA	4.32	50.74%	34.8	55.62%	84.7	62.66%	13.8	81.31%	20.3	71.72%
Bayesian	3.86	55.94%	34.2	56.36%	82.1	63.8%	12.9	82.52%	19.9	72.64%
PSO	3.84	56.22%	37.2	52.52%	80.5	64.5%	13.9	81.22%	20.2	71.95%
The Proposed Hyper-parameter Optimization method										
AGA	4.26	51.32%	36.7	53.24%	83.6	63.15%	13.4	81.91%	20.6	71.28%
GA+KNN	3.4	61.17%	30.11	61.6%	74.34	67.23%	12.46	83.13%	18.09	74.78%
Our method	3.43	60.89%	30.11	61.6%	74.75	67.05%	12.46	83.13%	18.17	74.67%
ES	3.37	61.53%	30.11	61.6%	73.74	67.5%	12.46	83.13%	17.94	75.03%

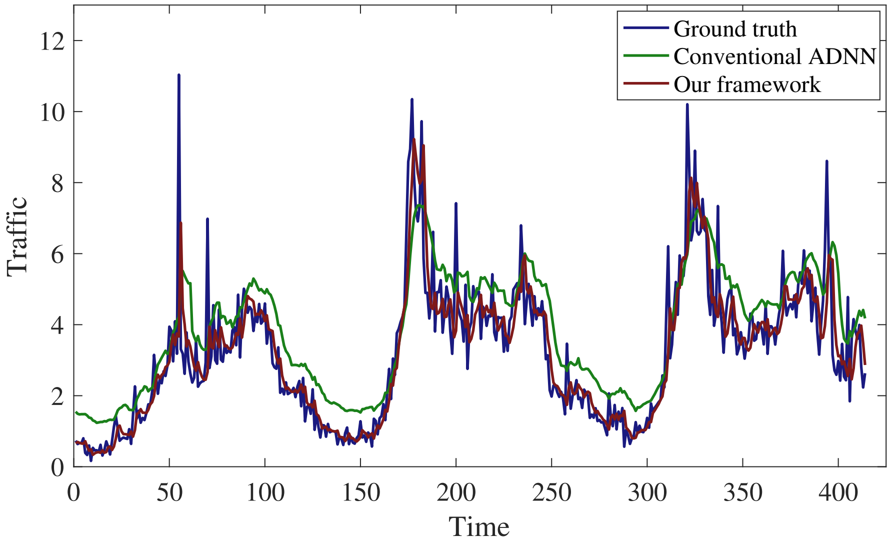


Fig. 5.13 Predicted traffic loads of the conventional ADNN and our framework as well as the ground true traffic loads generated in mobile cell 785.

and letting these proper hypothesis functions be efficiently found in the base-learner's training process.

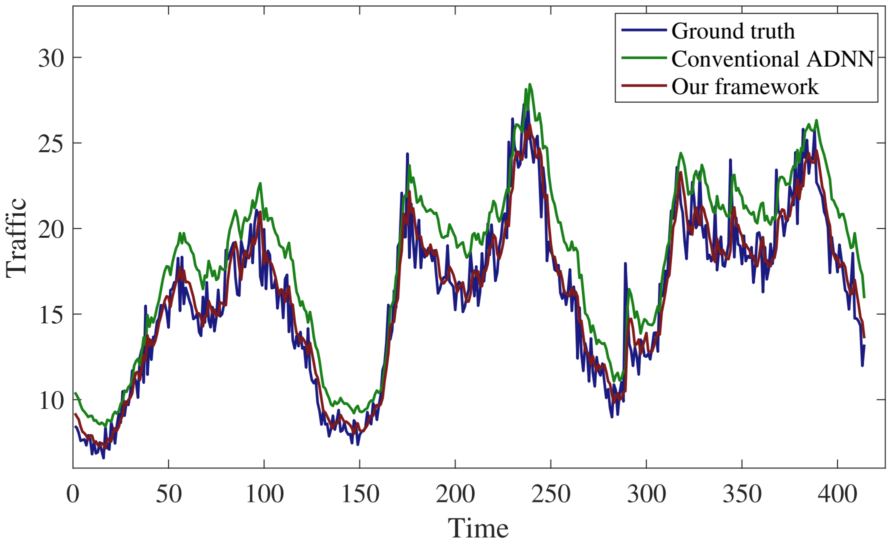


Fig. 5.14 Predicted traffic loads of the conventional ADNN and our framework as well as the ground true traffic loads generated in mobile cell 6708.

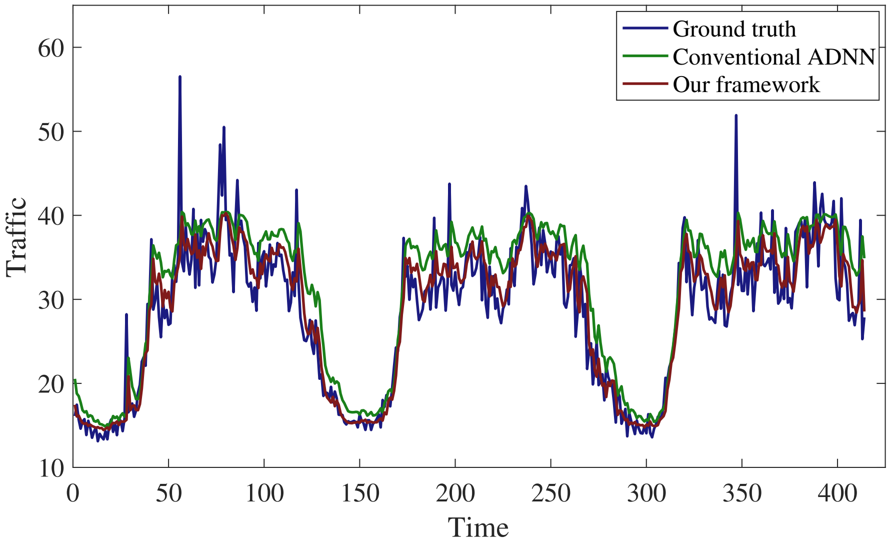


Fig. 5.15 Predicted traffic loads of the conventional ADNN and our framework as well as the ground true traffic loads generated in mobile cell 9106.

Compared with Bayesian optimization, PSO, GA, and AGA, meta-learner in the proposed framework can always obtain better hyper-parameter selection strategies for the testing base-learners. This can be attributed to the fact that the presented KNN learning algorithm in the meta-learner derives high-quality first-generation chromosomes, i.e., initial searching points of each hyper-parameter optimization problem, and makes it much easier for the following advanced genetic algorithm to find a satisfactory solution in the problem's huge solution space. From Table 5.2, we also see that hyper-parameters provided by GA+KNN or even ES can hardly further enhance the base-learners' after-training performance in comparison with our framework. These results demonstrate that the proposed meta-learner can efficiently generate appropriate solution for each hyper-parameter optimization problem which is very close to the theoretically optimal one, and also indicate that the adopted GRN based deep neural network in the advanced genetic algorithm can effectively screen out high-quality son chromosomes in each generation even though the practical fitness values of those son chromosomes are not calculated.

Figs. 5.13, 5.14, and 5.15, show the predicted traffic loads of the conventional ADNN and our proposed framework, as well as the ground true traffic loads generated in three randomly selected testing mobile cells, i.e., mobile cells 785, 6708, and 9106, respectively. It is obvious that the proposed framework can help the base-learners adapt to different base-tasks and predict cell-level wireless network traffic loads accurately supported by the suitable hyper-parameter selection strategies. On the other hand, the conventional ADNNs with randomly selected hyper-parameters for cells 785 and 6708 have considerable prediction errors even though they are well trained, which emphasizes the importance of hyper-parameter optimization for traffic prediction models and the importance of this research work.

5.4.5 Computational Time of the Proposed Framework

Fig. 5.16 shows the average on-line computational time of the considered hyper-parameter optimization methods, i.e., the proposed framework, GA, AGA, GA+KNN, and ES, over the testing base-tasks.

From Fig. 5.16, we can see that the ES consumes the most on-line computational time. This is because the ES must calculate the fitness value of each hyper-parameter selection strategy for the base-learner of every testing base-task and thus train the base-learner the same number of times as the amount of all the possible hyper-parameter selection strategies. Obviously, when the hyper-parameter optimization problem possesses a huge solution space,

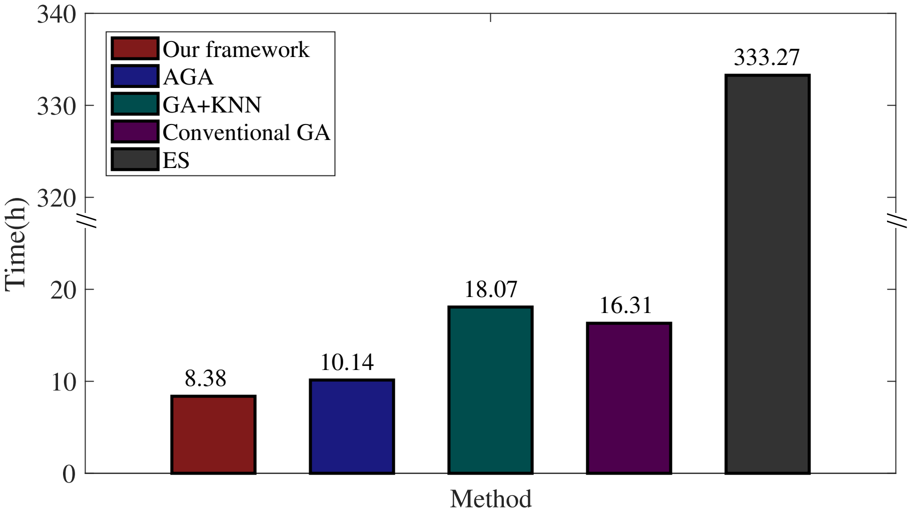


Fig. 5.16 The average on-line computational time of the considered hyper-parameter optimization methods.

the ES may not be practical due to its enormous computational complexity. Thanks to the novel deep learning assisted chromosome screening scheme, the proposed framework and AGA can quickly pick out the survived son chromosomes from all the generated ones without figuring out their exact fitness values in each iteration (chromosome generation). As a result, the proposed framework and AGA have much less on-line computational complexities compared with GA and GA+KNN. Conceivably, the computational complexity advantage of the proposed framework and AGA over GA and GA+KNN will further increase if the ratio between W and M augments. An interesting phenomenon we can also observe from Fig. 5.16 is that the proposed framework has smaller computational complexity than AGA in solving the hyper-parameter optimization problems even though they both adopt the intelligent chromosome screening scheme. This is because owing to the high-quality first-generation chromosomes provided by the KNN learning method, son chromosomes duplicated from the high-quality parent ones will have higher probabilities to survive at initial stage of **Algorithm 2**, avoiding considerable computational time in calculating these chromosomes' fitness values in the following iterations.

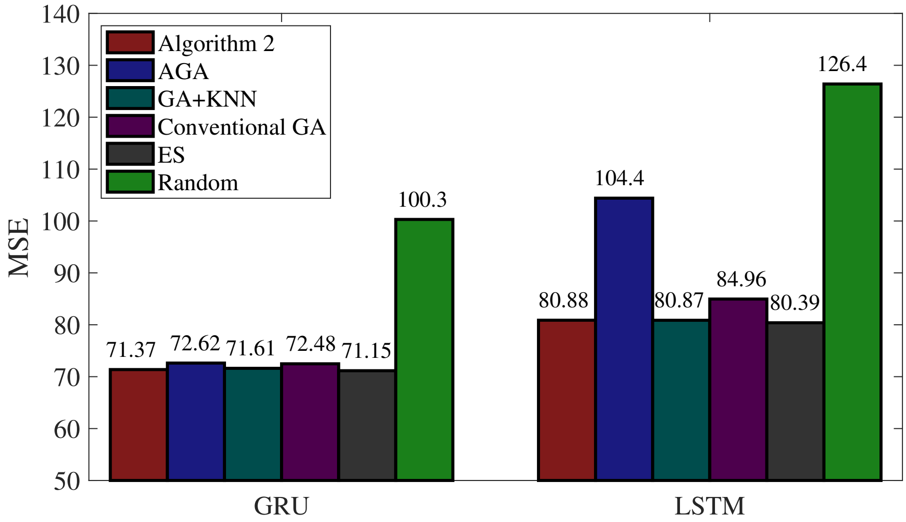


Fig. 5.17 MSE performance achieved by the considered hyper-parameter optimization methods when base-learners adopt other deep learning algorithms rather than ADNN.

Please note that the proposed framework needs relatively long computational time to construct the set of meta-samples for its meta-learner. However, since these meta-samples will be obtained off-line and the meta-knowledge only needs to be prepared once for the proposed framework, this extra off-line computational complexity can be justified by the improved after-training prediction accuracy of base-learners.

5.4.6 Robustness Analyses of Algorithm 2

We evaluate the robustness of **Algorithm 2** if base-learners adopt other deep learning-based models for cell-level wireless NTP.

Specifically, when each base-learner is designed as an LSTM network, or a GRU network, and the corresponding hyper-parameter selection ranges are listed in Table I, Fig. 5.17 and Fig. 5.18 show how the considered hyper-parameter optimization methods can elevate base-learners' after-training prediction accuracy in terms of the average MSE and R2 coefficient over the testing base-tasks by providing them proper hyper-parameter selection strategies. Fig. 5.19 demonstrates the on-line computational time of **Algorithm 2** in comparison to other hyper-parameter optimization methods. Please note that for each deep learning algorithm, we

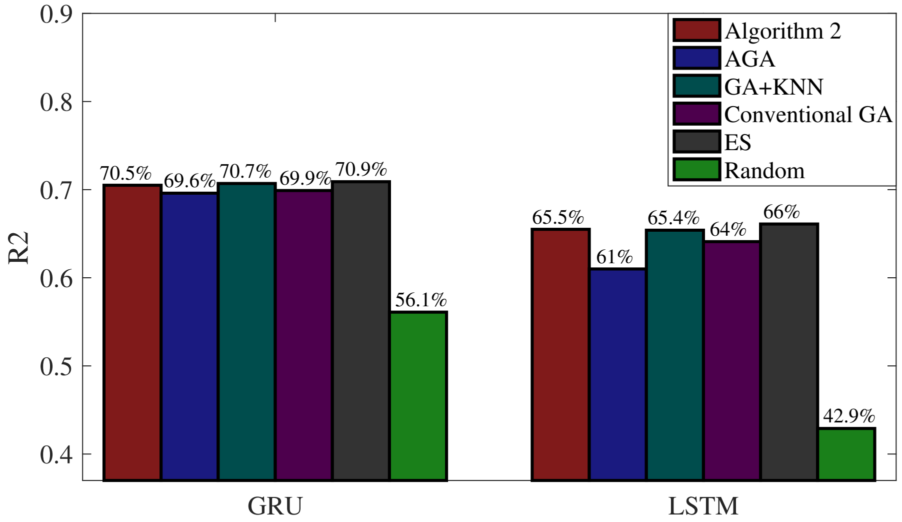


Fig. 5.18 R2 performance achieved by the considered hyper-parameter optimization methods when base-learners adopt other deep learning algorithms rather than ADNN.

construct identical set of meta-samples for **Algorithm 2** and each chromosome in **Algorithm 2** represents a feasible hyper-parameter selection strategy of the base-learner related to this deep learning algorithm. From these three figures, we can observe clearly that compared with randomly selected hyper-parameters, **Algorithm 2** will always significantly decrease base-learners' average MSE, increase base-learners' average R2, and possess acceptable on-line computational time no matter what deep learning algorithm these base-learners adopt. Moreover, compared with ES, **Algorithm 2** can always provide base-learners hyper-parameter selection strategies making them achieve after-training prediction accuracies very similar to those achieved with the theoretically optimal hyper-parameter selection strategies. Figs. 5.17, 5.18, and 5.19 demonstrate that **Algorithm 2** is robust to deep learning algorithms the base-learners are with.

5.5 Conclusion

In this chapter, a novel hyper-parameter optimization method has been proposed for cell-level wireless NTP with a meta-learning framework. In the framework, the ADNN is adopted as the base-learner to perform each cell-level wireless NTP task, i.e., base-task.

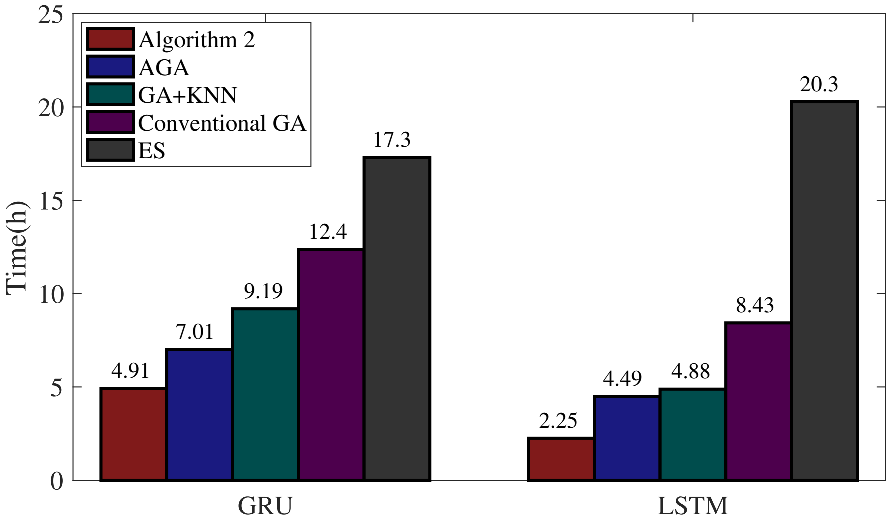


Fig. 5.19 The average on-line computational time of the considered hyper-parameter optimization methods when base-learners adopt other deep learning algorithms rather than ADNN.

What is more, we have defined finding the optimal hyper-parameters for the base-learner of each base-task according to the base-task's meta-features as the meta-task, and proposed a novel meta-learner to solve it. Based on our observation from real-world traffic records that base-tasks possessing similar meta-features tend to favour similar hyper-parameters for their base-learners, the meta-learner has exploited a KNN learning method to obtain a set of high-quality candidate hyper-parameter selection strategies for a new base-learner with the assistance of meta-knowledge and then utilized an advanced genetic algorithm with intelligent deep learning assisted chromosome screening to finally search the optimal solution of the hyper-parameter optimization problem. We have examined the performance of our framework through real-world cell-level wireless NTP tasks. Extensive experiments demonstrate that our framework can significantly elevate the base-learners' after-training prediction accuracy by providing them near optimal hyper-parameter selection strategies. Moreover, we have also revealed that the proposed meta-learner is robust to deep learning algorithms adopted by base-learners.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

This dissertation focuses on the NTP technology and considers the overall performance of the NTP model including prediction accuracy, computational efficiency and interpretability. In this dissertation, wireless network traffic has been divided into two categories, i.e. aggregate-level and cell-level wireless network traffic, based on whether or not overall user behavior patterns can be constructed from historical traffic data. Then we have researched the wireless network traffic patterns in various scenario, and proposed corresponding methods to construct specific NTP models. In comparison to benchmark methods, the proposed methods significantly improve overall prediction performance.

Chapter 3 and Chapter 4 focus on the aggregate-level wireless network traffic. The emphasis of Chapter 3 is regular aggregate-level wireless network traffic without special events. In Chapter 3, we have considered the connections between the user behavior and the corresponding wireless network traffic pattern, inventively utilized the user behavior into the construction of the NTP models, and laid the foundation of the following works in Chapter 4. In comparison with the benchmark models including the ARMA model, the ARIMA model, the MLP network, and the LSTM network, the proposed UBB NTP method in Chapter 3 significantly improves the interpretability and computational efficiency. In terms of prediction accuracy, the UBB NTP method and the LSTM network are comparable and far superior to other benchmark models. Furthermore, the standardized parameter set enables comparison of traffic patterns in different regions. Hence, this method is also valuable in the combination of communication and social science.

In Chapter 4, we have taken into account the nonroutine wireless network traffic which poses a severe challenge to existing NTP methods including the traditional statistics-based and the ML-based methods in terms of prediction accuracy, computational efficiency and interpretability. Thus, we have proposed the NNTP method in Chapter 4, and then constructed the SG-NNTP model as a case study of the NNTP method on the basis of real-world soccer-game events. Meanwhile, the corresponding single-step and multi-step prediction modes have been explained in detail. Similarly, the ARMA model, the ARIMA model, the MLP network, and the LSTM network are chosen as comparison models. By comparing the performance of the NNTP method with that of the benchmark methods, the NNTP method exhibits excellent results in prediction accuracy, computational efficiency and interpretability. Even in term of prediction accuracy where the deep learning-based NTP methods excel, the proposed NNTP method still achieves significant advantages.

Chapter 5 focuses on cell-level wireless network traffic data in which it is hard to extract the overall user behavior patterns due to its extreme complexities. We have chosen several models for cell-level wireless NTP tasks, including the ARIMA model, the SVR, the GP, the GRU network and the ADNN, and evaluated their prediction accuracy. Numerical results indicate that the deep learning-based models possess obvious advantages in comparison with the traditional statistics-based and the shallow learning-based models. Hence, in this context, deep learning-based models should be chosen to maximize the prediction accuracy. Furthermore, experiments in Chapter 5 have verified the huge impact of hyper-parameter selection strategies on these deep learning-based models. To develop the full potential of the deep learning-based models, we have investigated the hyper-parameter optimization problem and proposed a novel meta-learning based framework to solve the problem. In this framework, cell-level wireless NTP is regarded as the base-task corresponding a base-learner, and its hyper-parameter selection is the meta-task solved by a meta-learner. More specifically, the meta-learner utilizes the KNN algorithm to match a new base-learner with a set of high-quality hyper-parameter selection strategies which are then used in a GA algorithm as its initial chromosomes. To further improve the execution efficiency, we have introduced a GRN module to assist the chromosome screening. The experimental results in Chapter 5 shows that the proposed meta-learning based framework can significantly develop the base-learner's potential and improve its prediction accuracy with great execution efficiency. In addition, the robustness of the proposed meta-learning based framework has been validated as well.

6.2 Future Works

This thesis has investigated the aggregated-level NTP tasks and proposed interpretable NTP methods, i.e., the UBB NTP method and the NNTP method. For cell-level NTP tasks, we have proposed an attention-based deep neural network with a novel meta-learning based hyper-parameter optimization framework. These works can be extended in the following directions.

- Similar to the common challenge encountered in related works, most of the wireless network traffic data is regarded as sensitive data by the operators, and only very little and outdated wireless network traffic data is available to the public for academic research. In this thesis, the traffic datasets we used are those that took place in Milan in 2013 and those that took place in Guangzhou in 2017, which are outdated as well. In our future work, we will analyze the latest real-world wireless network traffic data from diverse regions with our interpretable methods.

- This thesis mainly focuses on the short-term trend of the wireless network traffic data, and has not taken the long-term trend and the seasonality into consideration, which is also due to the constrain of wireless network traffic datasets. The time span of the traffic datasets we used in this thesis ranges from a maximum of two months. In the future, we will investigate the long-term trend and the seasonality in the datasets with long time span to further enrich the proposed methods.

- As the preliminary research for NNTP, Chapter 4 just provides one case study. In the future, we will collect more nonroutine events and the corresponding traffic data. Subsequently, we will develop a method to capture the dependence between the events and the corresponding traffic data, and construct specific NNTP models for various nonroutine events. Finally, these NNTP models will be stored into a database, and a novel framework will be introduced to identify the nonroutine traffic and match it with suitable NNTP models. In addition, we will also consider the identification, classification and prediction of the nonroutine traffic in the absence of advance information.

- For cell-level wireless NTP model, our current proposed hyper-parameter optimization framework relies heavily on meta-knowledge. In other words, the meta-sample set must first be constructed. In future work, we will investigate whether the meta-task considered in this work can be solved with deep reinforcement learning technology in the absence of meta-knowledge.

References

[1] Ericsson, “Ericsson mobility report november 2023,” Nov. 2023, [online] Available: https://www.ericsson.com/4ae12c/assets/local/reports-papers/mobility-report/documents/2023/ericsson-mobility-report-november-2023.pdf.

[2] GSMA, “The mobile economy 2023,” 2023, [online] Available: https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2023/03/270223-The-Mobile-Economy-2023.pdf.

[3] C. Wang *et al.*, “On the road to 6G: visions, requirements, key technologies, and testbeds”, *IEEE Commun. Surv. Tutorials*, vol. 25, no. 2, pp. 905-974, Feb. 2023.

[4] W. Jiang, “Cellular traffic prediction with machine learning: A survey,” *Exp. Syst. Appl.*, vol. 201, Sep. 2022.

[5] G.O. Ferreira *et al.*, “Forecasting network traffic: a survey and tutorial with open-source comparative evaluation,” *IEEE Access*, vol. 11, pp. 6018-6044, Jan. 2023.

[6] I. Lohrasbinasab *et al.*, “From statistical- to machine learning-based network traffic prediction,” *Trans. Emerging Telecommun. Technol.*, vol. 33, no. 4, pp. e4394, Apr. 2022.

[7] A. D’ Alconzo *et al.*, “A survey on big data for network traffic monitoring and analysis,” *IEEE Trans. Network Serv. Manage.*, vol. 16, no. 3, pp. 800-813, Sep. 2019.

[8] R. Boutaba *et al.*, “A comprehensive survey on machine learning for networking: evolution, applications and research opportunities,” *J. Internet Serv. Appl.*, vol. 9, no. 1, pp.1-99, Dec. 2018.

[9] J. Hu *et al.*, “Base station sleeping mechanism based on traffic prediction in heterogeneous networks,” *ITNAC*, pp. 83-87, Nov. 2015.

[10] H. Hu *et al.*, “Computation offloading analysis in clustered fog radio access networks with repulsion,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10804-10819, Oct. 2021.

[11] L. Wang *et al.*, “Analytic network traffic prediction based on user behavior modeling”, *IEEE Networking Lett.*, vol. 5, no. 4, pp. 208-212, Dec. 2023.

[12] Z. Rao *et al.*, “Cellular traffic prediction: A deep learning method considering dynamic nonlocal spatial correlation, self-attention, and correlation of spatiotemporal feature fusion,” *IEEE Trans. Network Serv. Manage.*, vol. 20, no. 1, pp. 426-440, Mar. 2023.

[13] Y. Wang *et al.*, “Prediction of network traffic through light-weight machine learning,” *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1919-1933, Nov. 2020.

[14] R. Pascanu *et al.*, “On the difficulty of training recurrent neural networks,” *Proc. 30th Int. Conf. Machine Learning*, pp. 1310-1318, May 2013.

[15] M. Agiwal *et al.*, “Next generation 5G wireless networks: A comprehensive survey,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 1617-1655, Feb. 2016.

[16] Z. Zhang *et al.*, “dmTP: A deep meta-learning based framework for mobile traffic prediction,” *IEEE Wireless Commun.*, vol. 28, no. 5, pp. 110-117, Oct. 2021.

[17] J.M. Mendel *et al.*, “Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications,” *Proc. IEEE*, vol. 79, no. 3, pp. 278-305, Mar. 1991.

[18] J.M. Mendel *et al.*, “Short-term load forecasting via ARMA model identification including non-gaussian process considerations,” *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 673-679, May 2003.

[19] P. Chen *et al.*, “ARIMA-based time series model of stochastic wind power generation,” *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 667-676, May 2010.

[20] D. Zeng *et al.*, “Short term traffic flow prediction using hybrid ARIMA and ANN models,” *2008 Workshop Power Electron. Intell. Transp. Syst.*, pp. 621-625, Aug. 2008.

[21] A. Biernacki *et al.*, “Improving quality of adaptive video by traffic prediction with (F)ARIMA models,” *J. Commun. Networks*, vol. 19, no. 5, pp. 521-530, Oct. 2017.

[22] S. M. Tabatabaie Nezhad *et al.*, “A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks,” *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 700-703, Apr. 2016.

[23] M. Barabas *et al.*, “Evaluation of network traffic prediction based on neural networks with multi-task learning and multiresolution decomposition,” *2011 IEEE 7th Int. Conf. Intell. Comput. Commun. Process.*, pp. 95-102, Aug. 2011.

[24] A. Azari *et al.*, “Energy and resource efficiency by user traffic prediction and classification in cellular networks,” *IEEE Trans. Green Commun. Networking*, vol. 6, no. 2, pp. 1082-1095, Jun. 2022.

[25] W.S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115-133, Dec. 1943.

[26] B. Ding *et al.*, “Activation functions and their characteristics in deep neural networks,” *2018 Chin. Control Decis. Conf. (CCDC)*, pp. 1836-1841, Jun. 2018.

[27] M.M. Lau *et al.*, “Review of adaptive activation function in deep neural network,” *2018 IEEE-EMBS Conf. Biomed. Eng. Sci. (IECBES)*, pp. 686-690, Dec. 2018.

[28] M. Kaloev *et al.*, “Comparative analysis of activation functions used in the hidden layers of deep neural networks,” *2021 3rd Int. Congr. Hum.-Comput. Interact. Optim. Rob. Appl. (HORA)*, pp. 1-5, Jun. 2021.

[29] V. Nair *et al.*, “Rectified linear units improve restricted Boltzmann machines,” *Proc. 27th int. conf. Mach. Learn.*, pp. 807-814, 2010.

[30] V. Monga *et al.*, “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing,” *IEEE Signal Process Mag.*, vol. 38, no. 2, pp. 18-44, Mar. 2021.

[31] P.J. Werbos, “Backpropagation through time: What it does and how to do it,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550-1560, Oct. 1990.

[32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[33] S. Hochreiter *et al.*, “Gradient flow in recurrent nets: The difficulty of learning long-term dependencies,”, 2001.

[34] A. Graves *et al.*, “Speech recognition with deep recurrent neural networks,” *2013 IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 6645-6649, May 2013.

[35] Y. Fang *et al.*, “Traffic speed prediction based on LSTM-graph attention network (L-GAT),” *2021 4th Int. Conf. Adv. Electron. Mater. Comput. Software Eng. (AEMCSE)*, pp. 788-793, Mar. 2021.

[36] M. Bkassiny *et al.*, “A deep learning-based signal classification approach for spectrum sensing using long short-term memory (LSTM) networks,” *2022 6th Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. (ICITISEE)*, pp. 667-672, Dec. 2022.

[37] K. Greff *et al.*, “LSTM: A search space odyssey,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.

[38] M. Mohammadi *et al.*, “Deep learning for IoT big data and streaming analytics: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2923-2960, 2018.

[39] H. Jelodar *et al.*, “Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach,” *IEEE J. Biomed. Health. Inf.*, vol. 24, no. 10, pp. 2733-2742, Oct. 2020.

[40] N. Bui *et al.*, “A survey of anticipatory mobile networking: context-based classification, prediction methodologies, and optimization techniques,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 3, pp. 1790-1821, 2017.

[41] Y. Hua *et al.*, “Deep learning with long short-term memory for time series prediction,” *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114-119, Jun. 2019.

[42] X. Ge *et al.*, “A new prediction method of alpha-stable processes for self-similar traffic,” *IEEE Global Telecommun. Conf.*, pp. 675-679, Nov. 2004.

[43] L. Tang *et al.*, “ARMA-prediction-based online adaptive dynamic resource allocation in wireless virtualized network,” *IEEE Access*, vol. 7, pp. 130438-130450, Sep. 2019.

[44] Y. Shu *et al.*, “Wireless traffic modeling and prediction using seasonal ARIMA models,” *IEICE Trans. Commun.*, vol. 88, no. 10, pp. 3992-3999, Oct. 2005.

[45] F. Xu *et al.*, “Big data driven mobile traffic understanding and forecasting: A time series approach,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 5, pp. 796-805, Oct. 2016.

[46] N. C. Anand *et al.*, “GARCH — non-linear time series model for traffic modeling and prediction,” *NOMS 2008-2008 IEEE Network Oper. Manage. Symp.*, pp. 694-697, Apr. 2008.

[47] R. Li *et al.*, “The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice,” *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 234-240, Jun. 2014.

[48] F. Ju *et al.*, “Analysis of self-similar traffic based on the On/Off model,” *Int. Workshop Chaos-Fractals Theor. Appl.*, pp. 301-304, Nov. 2009.

[49] M. C. Falvo *et al.*, “Kalman filter for short-term load forecasting: an hourly predictor of municipal load,” *Proc. IASTED ASM*, pp. 364-369, Aug. 2007.

[50] X. Chen *et al.*, “Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale,” *2015 IEEE Int. Conf. Commun. (ICC)*, pp. 3585-3591, Jun. 2015.

[51] D. Tikunov *et al.*, “Traffic prediction for mobile network using Holt-Winter’s exponential smoothing,” *2007 15th Int. Conf. Software Telecommun. Comput. Networks*, pp. 1-5, Sep. 2007.

[52] Ankita *et al.*, “Comparative analysis of shallow learning and deep learning,” *2023 Int. Conf. Next Gener. Electron. (NEleX)*, pp. 1-8, Dec. 2023.

[53] N. I. Sapankevych *et al.*, “Time series prediction using support vector machines: A survey,” *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24-38, May 2009.

[54] J. Wang *et al.*, “User traffic collection and prediction in cellular networks: Architecture platform and case study,” *2014 4th IEEE Int. Conf. Network Infrastruct. Digital Content*, pp. 414-419, Sep. 2014.

[55] R. Han *et al.*, “Application of support vector machine to mobile communications in telephone traffic load of monthly busy hour prediction,” *2009 Fifth Int. Conf. Nat. Comput.*, pp. 349-353, Aug. 2009.

[56] Y. Zhang *et al.*, “SVR based voice traffic prediction incorporating impact from neighboring cells,” *2016 19th Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, pp. 142-146, Nov. 2016.

[57] S. Dawoud *et al.*, “Optimizing the power consumption of mobile networks based on traffic prediction,” *2014 IEEE 38th Annu. Comput. Software Appl. Conf.*, pp. 279-288, Jul. 2014.

[58] B. Ułanowicz *et al.*, “Combining random forest and linear regression to improve network traffic prediction,” *2023 23rd Int. Conf. Transparent Opt. Networks (ICTON)*, pp. 1-4, Jul. 2023.

[59] Y. Xu *et al.*, “Wireless traffic prediction with scalable Gaussian process: Framework algorithms and verification,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291-1306, Jun. 2019.

[60] X. Zhang *et al.*, “Adaptive gaussian process spectral kernel learning for 5G wireless traffic prediction,” *2022 IEEE 32nd Int. Workshop Mach. Learn. Signal Process. (MLSP)*, pp. 1-6, Aug. 2022.

[61] M. de los Ángeles Carrión-Herrera *et al.*, “Peak hour performance prediction based on machine learning for LTE mobile cellular network,” *2022 IEEE ANDESCON*, pp. 1-6, Nov. 2022.

[62] R. Holanda Filho *et al.*, “Network traffic prediction using PCA and K-means,” *Proc. IEEE Netw. Operations Manage. Symp. (NOMS)*, pp. 938-941, Apr. 2010.

[63] H. Drucker *et al.*, “Support vector regression machines,” *Proc. Int. Conf. Neural Inf. Process. Syst.*, pp. 155-161, 1997.

[64] L. Nie *et al.*, “Network traffic prediction based on deep belief network in wireless mesh backbone networks,” *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, pp. 1-5, Mar. 2017.

[65] Z. Wang *et al.*, “Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach,” *IEEE Trans. Ind. Inf.*, vol. 19, no. 4, pp. 5722-5731, Apr. 2023.

[66] J. Zhou *et al.*, “Multiscale network traffic prediction method based on deep echo-state network for internet of things,” *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21862-21874, Nov. 2022.

[67] H.D. Trinh *et al.*, “Mobile traffic prediction from raw data using LSTM networks,” *2018 IEEE 29th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, pp. 1827-1832, Sep. 2018.

[68] S. Jaffry *et al.*, “Cellular traffic prediction using recurrent neural networks,” *2020 IEEE 5th Int. Symp. Telecommun. Technol. (ISTT)*, pp. 94-98, Nov. 2020.

[69] Z. Wang *et al.*, “Data-augmentation-based cellular traffic prediction in edge-computing-enabled smart city,” *IEEE Trans. Ind. Inf.*, vol. 17, no. 6, pp. 4179-4187, Jun. 2021.

[70] J. Wang *et al.*, “Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach,” *IEEE INFOCOM 2017 - IEEE Conf. Comput. Commun.*, pp. 1-9, May 2017.

[71] C. Qiu *et al.*, “Spatio-temporal wireless traffic prediction with recurrent neural network,” *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554-557, Aug. 2018.

[72] Y. Hua *et al.*, “Traffic prediction based on random connectivity in deep learning with long short-term memory,” *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, pp. 1-6, Aug. 2018.

[73] S. Wang *et al.*, “A multitask learning-based network traffic prediction approach for SDN-enabled industrial internet of things,” *IEEE Trans. Ind. Inf.*, vol. 18, no. 11, pp. 7475-7483, Nov. 2022.

[74] X. Ma *et al.*, “Cellular network traffic prediction based on correlation convLSTM and self-attention network,” *IEEE Commun. Lett.*, vol. 27, no. 7, pp. 1909-1912, Jul. 2023.

[75] C. Zhang *et al.*, “Citywide cellular traffic prediction based on densely connected convolutional neural networks,” *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656-1659, Aug. 2018.

[76] Q. Zeng *et al.*, “Traffic prediction of wireless cellular networks based on deep transfer learning and cross-domain data,” *IEEE Access*, vol. 8, pp. 172387-172397, Sep. 2020.

[77] Z. Tian *et al.*, “A novel network traffic combination prediction model,” *Int. J. Commun. Syst.*, vol. 35, no. 7, pp. e5097, May 2022.

[78] Y. Hu *et al.*, “Citywide mobile traffic forecasting using spatial-temporal downsampling transformer neural networks,” *IEEE Trans. Network Serv. Manage.*, vol. 20, no. 1, pp. 152-165, Mar. 2023.

[79] Y. Yao *et al.*, “MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction,” *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 2837-2849, May 2023.

[80] F. Li *et al.*, “A meta-learning based framework for cell-level mobile network traffic prediction,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 4264-4280, Jun. 2023.

[81] S. Almeida *et al.*, “Spatial and temporal traffic distribution models for GSM,” *Proc. Gateway to 21st Century Commun. Village. VTC 1999-Fall. IEEE VTS 50th Veh. Technol. Conf. (Cat. No.99CH36324)*, pp. 131-135, Sep. 1999.

[82] I. Trestian *et al.*, “Measuring serendipity: connecting people, locations and interests in a mobile 3G network,” *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, pp. 267-279, Nov. 2009.

[83] M.R. Vieira *et al.*, “Characterizing dense urban areas from mobile phone-call data: Discovery and social dynamics,” *2010 IEEE Second Int. Conf. Social Comput.*, pp. 241-248, Aug. 2010.

[84] H. Chen *et al.*, “Online prediction algorithm of the news’ popularity for wireless cellular pushing,” *2015 IEEE/CIC Int. Conf. Commun. China (ICCC)*, pp. 1-5, Nov. 2015.

[85] K. Scout M. *et al.*, “I’ll work out tomorrow: the procrastination in exercise scale,” *J. Health Psychol.*, vol. 26, no. 13, pp. 2613-2615, Nov. 2021.

[86] K. Katrin B. *et al.*, “Procrastination: when good things don’t come to those who wait,” *Eur. Psychologist*, vol. 18, no. 1, pp. 24-34, 2013.

[87] Z. Chen *et al.*, “Neural markers of procrastination in white matter microstructures and networks,” *Psychophysiology*, vol. 58, no. 5, pp. e13782, Jan. 2021.

[88] G. Barlacchi *et al.*, “A multi-source dataset of urban life in the city of Milan and the Province of Trentino,” *Sci. Data*, vol. 2, no. 1, pp. 1-15, Apr. 2015.

[89] G. Costa *et al.*, “City report: Milan,” *WILCO Publ.*, vol. 23, 2012.

[90] M. Niu *et al.*, “Financing urban growth in China: A case study of Guangzhou,” *Aust. J. Social Issues*, vol. 55, no. 2, pp. 141-161, Jun. 2020.

[91] Y. Wang *et al.*, “An integrated framework for service quality, customer value, satisfaction: Evidence from China’s telecommunication industry,” *Inf. Syst. Front.*, vol. 6, no. 4, pp. 325-340, Dec. 2004.

[92] D. Jhamb *et al.*, “The behavioural consequences of perceived service quality: A study of the Indian telecommunication industry,” *Bus.: Theory Pract.*, vol. 21, no. 1, pp. 360-372, May 2020.

[93] S. Samulevicius *et al.*, “MOST: Mobile broadband network optimization using planned spatio-temporal events,” *2015 IEEE 81st Veh. Technol. Conf. (VTC Spring)*, pp. 1-5, May 2015.

[94] F. Botta *et al.*, “Quantifying crowd size with mobile phone and Twitter data,” *R. Soc. Open Sci.*, vol. 2, no. 5, May 2015.

[95] G. Li *et al.*, “Estimate the limit of predictability in short-term traffic forecasting: An entropy-based approach,” *Transp. Res. Part C Emerging Technol.*, vol. 138, May 2022.

[96] C. Lemke *et al.*, “Meta-learning: A survey of trends and technologies,” *Artif. Intell. Rev.*, vol.44, pp. 117-130, 2015.

[97] A. Vaswani *et al.*, “Attention is all you need,” *Proc. Adv. Neural Inf. Process. Syst.*, pp. 5998-6008, 2017.

[98] A. Lambora *et al.*, “Genetic algorithm- A literature review,” *Proc. Int. Conf. Mach. Learn. Big Data Cloud Parallel Comput. (COMITCon)*, pp. 380-384, Oct. 2019.

[99] K. Tan *et al.*, “Gated residual networks with dilated convolutions for supervised speech separation,” *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 21-25, Sep. 2018.

[100] C. Zhang *et al.*, “Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389-1401, Jun. 2019.

[101] I. Loshchilov *et al.*, “Decoupled weight decay regularization,” in arXiv: 1711.05101, 2017.