Machine Learning Based Mobile Network Data Analysis and Prediction in Wireless Communication Network

Zhezi Zhao

Department of Electronic and Electrical Engineering

University of Sheffield

Supervisors: Prof. Jie Zhang, Dr. Hualiang Wei

This thesis is submitted for the approval of the

Doctor of Philosophy

July 2023

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

Acknowledgements

First and foremost, I would like to express my sincere thanks to my supervisor Prof. Jie Zhang for the continuous guide and support during my PhD study. His patience, encouragement and guidance help me to finish my study. His optimism and confidence towards life and work have been a huge inspiration to me. Besides my supervisor, I would like to thank my second supervisor, Dr. Hua-Liang Wei, who gives me great support.

I am truly grateful for Dr. Zitian Zhang, an outstanding scholar who gives me continuous encouragement, guidance, and support. His creative thinking and passion for research have been a constant source of inspiration, motivating me to become a better researcher.

I would also like to extend my heartfelt gratitude to my colleagues, Dr. Bo Ma and Dr. Fuyou Li. Their support and companionship during our days and nights in the laboratory, as well as in various aspects of life, have given me a strong sense of belonging and encouragement to persist in my research.

Finally, I would like to say thank you to my parents and my aunt for their selfless care and companionship in my life. Their unwavering support and encouragement have always been the driving force behind my perseverance. “XIEXIE!”

Abstract

With the rapid development of wireless networks, more and more online services significantly raise mobile data traffic demands, which causes a massive challenge for wireless network operators. In addition to deploying more communication facilities to improve the whole wireless network capacity, real-time observation and prediction of the mobile data traffic to achieve a dynamic balance of network load can further improve the efficiency of the services from those operators and reduce energy waste. In general, the accuracy of mobile data traffic prediction directly impacts the entire network system's Quality of Service (QoS) and Operating Expenditure (OPEX). Therefore, mobile network traffic prediction is the main research direction of this thesis.

Firstly, the user's Point of Interest (POI) exploration is chosen as a key point for analysis. This kind of user mobility modelling represents an essential branch of mobile traffic analysis. By applying machine learning algorithms, clear summaries of the mobility pattern characteristics of typical wireless users are obtained. Through the analysis of these regular characteristics, the value of mobility information related to the user's POI is initially demonstrated. Subsequently, this thesis introduces a data prediction model based on the Long Short-Term Memory (LSTM) model, a typical neural network for sequence prediction. Through verification of predictions using real sampled user data, it further demonstrates that the user's POIs tend to be relatively fixed and exhibit periodicity. Additionally, by comparing the prediction results with those of other models, the advantages of neural networks, particularly LSTM, in sequence prediction are evident.

Subsequently, this thesis aims to enhance the accuracy of wireless data traffic prediction by exploring location information. Although former researches have indicated that the distance relationship may affect the similarity of mobile communication traffic across different base stations, there is a lack of studies regarding the selection of the training dataset scope for urban mobile traffic. Building upon the previous research on the user's POI characteristics, this thesis verifies that the mobile network data trends of base stations in distant regions could also exhibit high similarity with applying real-world data, thereby expanding the training sample range. After that, a multi-task learning framework called MTL-STPN is designed to incorporate these highly correlated mobile traffic data as auxiliary content for predicting target region mobile traffic data. The results demonstrate that the designed model achieves nearly a 10% improvement in mobile traffic prediction compared to the state-of-the-art traffic prediction models with Root Mean Square Error (RMSE) measurement as prediction metrics. This outcome substantiates that reasonable correlations between mobile network traffic samples can be applied to enhance the performance of appropriate algorithms.

Finally, to address the more complex and bursty but highly valuable application-level mobile network traffic prediction, specifically Instant Messaging (IM), this thesis further improves upon the characteristics of the sub-models extracted from the multi-task framework and proposes a novel deep stacked learning architecture called SLIM-TP. After operating the sub-models for extracting the spatiotemporal dependencies of traffic as well as the mobile users' equipment (UEs) behavioural information, the meta-learner is employed to make optimal decisions regarding these features and effectively retain the factors that can enhance prediction accuracy. Experimental results based on a large dataset collected from a real cellular network demonstrate that the proposed model achieves over 40% improvements in WeChat traffic prediction performance compared to the state-of-the-art traffic prediction models through RMSE measuring. It shows the effectiveness of incorporating high-dimensional data such as user location and related traffic as auxiliary features in complex mobile network traffic prediction scenarios.

List of Abbreviations

3GPP	3rd Generation Partnership Project
5G	the Fifth Generation
ACF	Autocorrelation Function
AIS	Automatic Identification System
ANN	Artificial Neural Networks
AP	Access Point
API	Application Programming Interface
ARIMA	AutoRegressive Integrated Moving Average
BS	Base Station
CEs	Conditional Entropy
CNN	Convolutional Neural Networks
CpnvLSTM	Convolutional Long Short-Term Memory
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DFT	Discrete Fourier Transform

FFT	Fast Fourier Transform
FPMC	Factorising Personalised Markov Chain
FT	Fourier Transform
GP	Gaussian Process
GPS	GPS Global Positioning System
KDE	Kernel Density Estimation
KPI	Key Performance Indicator
LR	Linear Regression
LSTM	Long Short-Term Memory
MC	Markov Chains
MEC	Multi-Access Edge Computing
MinPts	Minimum number of points
MLP	Multilayer Perceptron
MTL	Multi-task Learning
MTL-STPN	Multi-Task Learning-based Spatial-Temporal Parallel deep learning Network
NLP	Natural Language Processing
NN	Neural Network
OPEX	Operating Expenditure
POI	Point of Interest
PPM	Periodic Pattern Mining

QoS	Quality of Service
R2	R-Squared
RAN	Radio Access Network
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
RRM	Radio Resource Management
RSSI	Received Signal Strength Indication
SLIM-TP	Stack Learning-based framework for Instant Message Traffic Prediction
SON	Self-Organising Network
SSE	Sum of Squared Errors
SVR	Support Vector Regression
UAV	Unmanned Aerial Vehicle
UE	user equipment
UGC	User Generated Content

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background and Motivation

This section provides an overview of the current state of mobile network traffic prediction from three perspectives. Firstly, the concept of traffic pattern analysis is proposed to explain the feasibility of modelling and predicting cellular network traffic. Next, various types of data related to cellular network traffic are listed to illustrate their different characteristics and applications in mobile traffic prediction. Finally, the machine learning algorithms used in traffic prediction are introduced to highlight the challenges faced in practical implementations.

1.1.1 Mobile traffic analysis and prediction

Mobile traffic analysis has emerged as one of the most popular research fields in recent years [1], starting from the analysis of large-scale mobile network traffic datasets since 2006 [2]. With the technical advancement of the cellular network, research based on network traffic data has benefitted various domains. These research efforts can be broadly categorised into three main themes based on their focal point, which are network analysis, mobility analysis, and social analysis [65]. In network analysis, two aspects are mainly studied, one of which is to understand the dynamic trend of mobile traffic, and another one is to update the configuration strategy in the mobile network to adapt to the changes of various traffic patterns.

Many studies [66] have proved that modelling and predicting mobile traffic fluctuations can greatly reduce OPEX and increase network capacity. Since reducing OPEX and increasing network capacity are very valuable in terms of practical operation, the research on mobile traffic prediction becomes critical. The research objects of mobility analysis are individuals, crowds, and transportation, which is reasonable since mobile traffic is generated by user behaviour. User crowds with different attributes have various wireless network data usage habits, resulting in diverse traffic patterns. Therefore, the data of UEs' behaviour such as trajectory, user habit, and event affected can be used to assist in understanding and modelling traffic trends. At present, the historical trajectories can be recorded through the handover log from the BS or the location information from the local devices [67], which are desensitised based on the concern of security and concealment. Furthermore, compared to the analysis of network traffic and user mobility pattern, the relationship and interaction between the UEs are the focused research direction in the study of the social analysis aspect. Besides, as aforementioned, event detection is another direction in social analysis, which is mainly used in anomaly detection of mobile network traffic. Therefore, the analysis of wireless network traffic could be based on multi-dimensional information related to the network data flow, which is helpful to understand and predict traffic patterns.

Among the various topics of network analysis, mobile network traffic prediction holds particular significance. Numerous types of research rely on predicting traffic in advance, which encompasses network optimisation aspects [3] such as resource deployment, load balancing, caching, hotspot/blackspot detection, and base station sleep strategies. These studies emphasise that the ability to anticipate adjustments to infrastructure performance can have a significant impact on system operational efficiency and energy utilisation efficiency [3]. In other words, the research of mobile traffic analysis and prediction is to enable the above strategies to be executed earlier to offset the impact of mobile traffic fluctuations. Therefore, there are two aspects of mobile traffic prediction that have been extensively studied. One is to improve the accuracy of the prediction, so as to make it more reliable for early deployment. The other is to improve forecasting efficiency and shorten forecasting time. To improve the accuracy, it is necessary to improve the completeness of the data set and the capabilities of feature extraction of the prediction model. While improving efficiency, it is necessary to clean the data and simplify the model as much as possible. These aspects together will form a trade-off strategy, whose parameters need to be carefully adjusted according to the different purposes and environments.

1.1.2 Data type in mobile traffic analysis and prediction

In mobile traffic analysis, different types of data need to be collected through the nodes, sensors, and devices in the heterogeneous network [4]. Statistics and analysis are performed according to the attributes, patterns, and objects of each type of data to obtain effective information. In terms of network optimisation, the most intuitive and common analysis is to use aggregate traffic flow data. This kind of data represents the trend of the traffic in the corresponding range, which is the Key Performance Indicator (KPI) that can directly represent the current status of the network. According to the location of the recorded data, aggregate traffic can usually be divided into network-level, cell-level, and application-level [5]. The aggregate traffic at the network-level refers to the network traffic of the core network. When it comes to mobile network traffic, it mainly focuses on cell-level and application-level traffic data obtained on the Radio Access Network (RAN) side. At cell-level, mobile traffic data is collected according to temporal (time interval) and spatial (BS, grid of the map), respectively. Such data usually have strong seasonality in general and are more suitable for modelling traffic in terms of temporal dependencies, so as to predict the traffic in the next timestamp [70]. The data is usually recorded in the traffic volume (including download and upload) at each time interval. Moreover, because the cell-level data is obtained from the BS side, combined with the geographic location of the BS or Point of Interest (POI), it is also possible to explore spatial features through the traffic pattern [71]. Application-level traffic is collected by the BS side as well. Unlike cell-level aggregate mobile network traffic, it can be subdivided by data application types and then can also be expanded into the dimensions of the spatial-temporal attributes. In this way, under the situation with the same time and location, different applications have different traffic patterns, which is helpful for the study of user behaviour. On the other hand, due to the nature of the application, the type of data generated from these applications are vary. For example, YouTube, Spotify, and Gmail mainly generate video, music, and text information respectively, which makes the amount of data used per unit time different. It is an essential feature in solving the problem of regional classification [72].

At the same time, due to the different usage scenarios of different applications, their active time will vary as well. For example, it is shown that IM applications have become one of the applications that occupy the most time and frequency of the users [73]. Therefore, apart from the research on the data volume, it can also be used in the study on the daily behaviour patterns of users. It is worth noting that as the granularity increases, traffic volume is counted in the increasingly smaller areas, which may result in a decrease in the number of users and hence increasing randomness of the traffic, especially for the more detailed application-level datasets. Therefore, how to extract useful features for network traffic modelling will become a critical research topic.

In addition to considering traffic volume datasets, heterogeneous datasets will be used as auxiliary information in network traffic forecasting as well. Among them, the most important one is grabbing geographic location information, such as Global Positioning System (GPS), which provides better accuracy of the location than the location information provided by BS. This is helpful for user-related research such as user mobility, POI [74] and urban transportation [75], and thus feedback on user behaviour patterns for the network optimisation specifically [3]. Another type of data that can be used in mobile traffic prediction is online data, which is the public data obtained through the Application Programming Interface (API) of the applications. It may include the data of geographic tags, user activity, event alarms, etc. For example, in [64], the author proved that there is a positive correlation between the social media data (Twitter), the network traffic, and the number of users, which then helps mobile operators to have new auxiliary information to predict network traffic.

From the above description that explains how to assist traffic forecasting through heterogeneous datasets, it can be seen that crowd behaviours are widely considered in the study of mobile traffic prediction. It is also reasonable to propose that the population and network load are related to each other, as a result, the mobile traffic analysis is also commonly used in population estimation [74]–[78].

To conclude, the studies on cell-level aggregate traffic, application-level traffic, geographic information capturing, and population distribution play important roles in the research on mobile traffic analysis and prediction.

1.1.3 Machine learning in mobile traffic forecasting

There are more and more researches that began to follow closely on the deep learning algorithms in data traffic forecasting. Among the algorithms, the most widely used ones are the Recurrent Neural Network (RNN) and its specific variants. The reason why RNN has an impressive performance in sequence prediction is that it can process data according to sequence changes [79], while the other neural networks are not cable to do so. Moreover, RNN can use the internal state to store and process sequences. It does not mean that RNN does not have any disadvantages. In fact, the most concerned one is the gradient vanishing and exploding problem. Therefore, LSTM, as a modified variant of RNN, are designed to avoid the long sequence dependency issue [80]. The long-term memory of LSTM makes this class of algorithms very effective. In conclusion, LSTM is an efficient method to train time series data and extract temporal features.

In order to further improve the prediction accuracy, spatiotemporal dependencies are required to be extracted, which requires data structure to change from being sequential to a matrix, and hence convolutional networks can be applied. Convolutional networks were initially used for image or video action recognition in deep learning applications. When the mobile network traffic heat-map of each timestamp in multi-cell is regarded as an object and analysed in the time domain, the periodic trend of mobile network traffic can be modelled for the purpose of prediction. Therefore, the convolution network algorithm combined with the sequence modelling method is widely proposed in network traffic forecasting. The 3D-CNN and Convolution-LSTM (ConvLSTM) are therefore being widely used [71]. Compared with CNN, 3D-CNN directly extracts temporal features by increasing the time dimension, but because of its natural characteristic, the results are only ideal for short-term prediction.

Compared to that, ConvLSTM has a long sequential memory ability based on its LSTM units which means it can accept spatial features by turning internal operations into convolution operations, but the cost is larger computation compared to the former model. Meanwhile, by proposing novel architectures, such as multi-task learning (MTL) [81] and transfer learning [71], the training efficiency can be increased and the prediction accuracy can be further improved as well by treating each sequence prediction as a sub-task.

To conclude, the application of deep learning is currently more popular in traffic prediction research because of its own specific feature capture ability and the ability to increase feature dimensions through architecture changes.

1.1.4 Motivation

In recent years, the challenges posed by the increasing demand for mobile network traffic have become extremely severe. According to a white paper published by Cisco, it is estimated that by 2022, the total monthly mobile data traffic will reach 77EB, which is seven times higher than the traffic monthly consumption in 2017 [62]. With the explosive increase of mobile devices and advanced multimedia functionalities, the number of mobile devices and cellular network connections is expected to grow by nearly 60% from 2018 to 2023 [63]. It indicates that significant load pressures may arise at any Access Point (AP), such as Base Stations (BS), which is necessary to allocate more network resources during peak times. Therefore, for network operators, balancing OPEX and QoS represents a challenging task.

In addition to upgrading and deploying mobile network infrastructure, analysing the spatiotemporal characteristics of network traffic to predict mobile network traffic trends has become another popular direction for increasing mobile network capacity [82]. With the widespread application of Fifth Generation (5G) mobile networks, cellular networks need to support capacity increases ranging from 600 to 2500 times [66]. The solution lies in the dense deployment of small cells, which is an effective approach to address current limitations in coverage, capacity, and traffic demands. However, such large-scale small cell deployments can result in significant energy consumption. The current approach of base station sleep strategies based on local traffic prediction is a common and effective solution to mitigate this issue. Moreover, network congestion and resource allocation problems are increasingly prominent. The Third Generation Partnership Project (3GPP) defines Self-Organising Networks (SON) as a network that can achieve most functionalities with little or no manual operation, enabling self-optimisation and self-healing [69] to address these issues. It means that all these solutions rely on reliable predictions of traffic trends to accomplish network optimisation. Furthermore, network assistance or emergency functions, such as base station power [6] or drone base stations [7], also require mobile traffic prediction and anomaly detection for resource deployment within a region. Therefore, mobile network traffic prediction has become a key probe for optimising the QoS of this expanding large cellular data network. It is the reason that it has gained significant value in mobile network optimisation in both academia and industry in recent years.

In addition, with the proliferation of mobile devices, applications, and small cells, mobile network traffic is becoming complex. According to statistics, there has been explosive growth in the variety of software types [8], including IM, online shopping, online gaming, video/image streaming, etc.. Most of these applications heavily rely on real-time data throughput. Each application category generates traffic with different patterns. Compared to traditional aggregated traffic, application-level traffic is more bursty and highly geographic-dependent, making it more challenging to predict. However, the analysis and prediction of such data are indeed becoming crucial for wireless network operators. For example, application-level traffic can be analysed to identify location attributions [9]. Therefore, the prediction of application-level data needs to be carefully considered. On the other hand, with the extensive deployment of small base stations, the traffic of each small base station also needs to be predicted. However, due to the limited monitoring area of each small base station, the user movements may be more frequent and random, resulting in more unstable traffic tendency compared to aggregated traffic. Meanwhile, the increasing number of small cells means more data to be predicted. Therefore, improving prediction efficiency while ensuring accuracy is a vital challenge in mobile traffic prediction topics.

Besides, the correlation of mobile network traffic data has gained increasing attention as well, particularly in terms of spatiotemporal correlation. The analysis of mobile network traffic initially focused on temporal periodicity, which remains a fundamental aspect in most traffic prediction approaches[10]. Recently, the spatial correlation has also been widely discussed [11], primarily studying the traffic characteristics around the target area and attempting to extract features using neural networks [12]. However, in addition to traffic, mobile networks also generate information about UEs, such as their locations and population densities. Current research on this type of data mainly focuses on exploring its inherent characteristics, such as user POI extraction [13], trajectory prediction [14], and estimation of epidemic outbreaks [15][16]. Back to mobile networks, user behaviour directly influences traffic variations. Research has shown that changes in traffic patterns can be partially attributed to the specific time and location of individual users [17][18]. For example, the traffic patterns of office employees differ during working hours and non-working hours, and the traffic patterns in schools in the afternoon may differ from those in central business districts. It is an interesting connection, indicating that information about future traffic may not only be derived from historical traffic data but also influenced by other user behaviours.

In terms of prediction models, several mature algorithms have been introduced, primarily divided into statistical-based models and machine learning-based models. Statistical-based models aim to describe the characteristics of mobile network traffic, mainly in periodicity and correlation, and use mathematical models to model these traffic features. Popular models in this category include Markov modelscite [19], α-stable modelscite [20], and autoregressive moving average models[21]. However, with increasing observation time and the developing more complex mobile network, these models face challenges in effectively and promptly addressing the current traffic prediction problems. As a result, machine learning-based models have gradually been applied in prediction models. These models are more flexible and avoid requiring the establishment of complex mathematical models. Traditional machine learning-based models, such as linear regression (LR) [135] and support vector regression (SVR) [136], can directly predict data with strong periodicity without the need for data preprocessing and modelling. In more complex scenarios, neural network-based models, such as Artificial Neural Networks (ANN), RNN, and even deep neural network models like Multilayer LSTM, can better fit the data and update traffic features more rapidly [21].

However, such models rely on large training datasets and long-term training [21]. Due to limited resources, deeper neural network models or larger training datasets are impractical as the demand for prediction accuracy increases and more complicated network traffic patterns. Therefore, there is a trade-off between model complexity and practical constraints, and finding the right balance between prediction accuracy and resource requirements remains a challenge in the field of mobile traffic prediction.

In summary, mobile network traffic prediction needs to deal with a range of challenges. It is important to consider the limitations of current research, as they provide directions for addressing these challenges. The following are some crucial challenges and limitations summarized in this thesis:

- The accuracy of mobile network traffic prediction is of paramount importance as it directly affects the management and operational efficiency of the entire wireless network.

- Currently, mobile network traffic patterns are becoming increasingly complex as the granularity rises, which calls for more precise models to accurately predict these patterns.

- Currently, the inherent characteristics of mobile network traffic and other correlated information have not been thoroughly extracted, which results in missing content and user behaviour. It is necessary to study the relevant data generated within the mobile network, as well as user behaviour.

- Although deep neural network models have gained popularity in effectively extracting pattern features and correlations, they operate with costs in terms of computational efficiency and training data volume. Therefore, the efficiency of prediction models needs to be seriously considered.

To address the challenges, this thesis focuses on analysing the characteristics and correlation of the traffic data and applying them to the proposed deep-learning-based mobile traffic prediction frameworks to enhance prediction performance.

1.2 Contributions of the Thesis

Based on the mobile traffic prediction, the main contributions of this thesis are listed as follows:

- Proposing a POI preference method to enhance the performance of user's next destination prediction. It applies clustering algorithms and periodic behaviour detection to construct individual Points of Interest (POI) based on user behaviour. By leveraging advanced neural network learning models, the result validates the patterns of mobile user locations, thereby determining the behavioural characteristics of mobile network users.

- Developing an effective multi-task learning-based mobile traffic prediction framework enables improved prediction of small area traffic. The proposed framework is validated using a series of accuracy testing algorithms. Additionally, the framework significantly enhances training time and computational efficiency.

- Based on the framework, a high-correlation data search algorithm is proposed. The algorithm effectively pre-processes large-scale mobile network traffic data, reducing the training data size and objectively improving the efficiency of the algorithm.

- Based on the excellent characteristics of the multi-task learning framework and the summarisation of user location features, a mobile network traffic prediction stacked learning framework is further proposed. This framework incorporates the extraction of multidimensional data features, where the weights of multiple learners are increased to extract corresponding data features. The framework also includes a meta-learner, a Multi-Layer Perceptron (MLP), for further feature selection. It not only models feature for mobile network traffic and user position information but also improves prediction accuracy. The proposed framework exhibits outstanding performance in forecasting highly dynamic application-level mobile network traffic.

- The designed algorithms are applied and tested using real-world mobile network traffic datasets. The practical application validates the potential of the algorithm in solving real-world problems and verifies its respective advantages.

1.3 Structure of the Thesis

The structure of this thesis is introduced as follows:

Chapter 2: Literature review
This chapter has introduced the development and existing work of mobile network traffic prediction. Then, machine learning techniques applied in mobile network prediction are discussed.

Chapter 3: Personal POI detection based on periodic pattern mining

In this chapter, a periodic pattern detection algorithm for geographic information is proposed to uncover the behavioural characteristics of users in the mobile network. The Periodic Pattern Mining (PPM) algorithm, which is primarily based on spatiotemporal information, has been widely used for user trajectory prediction. Simultaneously, map-based POI is no longer limited to providing map-related extension information in the field of network traffic prediction but should be considered as the subject of user-based research to establish user location preferences. In this chapter, Bayesian statistical methods are employed to establish user POI preferences. POI preferences involve classifying the user's historical geographic location information based on the information they most frequently visit and outputting the corresponding likelihood. After that, widely used sequence prediction models are applied to demonstrate the effectiveness of the proposed method for establishing individual preferences. Through this algorithm, the periodic visiting characteristics of users to specific areas are discovered and determined.

Chapter 4: MTL-STPN: a Multi-Task Learning-based Spatial-Temporal Parallel deep learning Network

In this chapter, a parallel-based framework for mobile network prediction is proposed. Existing researches have shown that extracting spatiotemporal features cannot increase efficiency as the target area expands. Therefore, it is worth investigating how to purposefully select areas and use more efficient frameworks for modelling. Multi-task learning is a challenging algorithm that involves multiple inputs and outputs and shares part of parameters in modelling to improve generalisation performance, making it more suitable for the current problem. Therefore, a deep learning model based on a parallel structure is proposed, which applies a multi-task learning architecture to optimise prediction performance. Additionally, a matching algorithm for detecting relevant areas is proposed based on the characteristic deduced in Chapter 3, which can select the network traffic of the alternative region with efficient feature information, thereby further improving accuracy while keeping the system complexity low.

Chapter 5: SLIM-TP: a deep Stack Learning-based framework for Instant Message Traffic Prediction

In this chapter, an advanced and fine-grained mobile network traffic prediction framework is proposed. Due to the different usage habits of users for different applications, the traffic patterns at the application level are various, especially for IM traffic. Modelling such patterns with existing aggregated mobile traffic models is quite challenging due to the small observation areas and high mobility of individuals. Therefore, to deal with highly bursty patterns, it is necessary to introduce more data to improve the efficiency of feature capture. Based on the characteristics of IM applications, in addition to basic temporal series prediction, spatiotemporal correlation models and local population distribution are also introduced for relevant modelling. Furthermore, following the characteristics of the multi-task learning framework in Chapter 4, this chapter further enhances the parallel learning module for different feature extraction. A meta-learner is also incorporated as the final feature selection and integration, integrating information extracted from each dimension of the data to accomplish IM traffic prediction. Additionally, the performance of the proposed framework in predicting mobile network traffic at other scales is also validated.

Chapter 6: Conclusion & Future work

This chapter provides a summary of the work presented in this thesis and outlines potential future directions for research.

Chapter 2

Literature Review

2.1 Mobile Network Traffic Prediction

The essence of mobile network traffic prediction is the analysis and feature extraction of the data flow transmitted in cellular networks. Such data is encapsulated as network packets and transmitted through wireless radio carriers [21]. It is precisely this physical characteristic that allows terminals to record the amount of data transmitted within a certain period of time, thereby understanding the network load [22]. Therefore, research on mobile network traffic can intuitively reflect the transmission quality of network channels to enable management and control [22]. This is the fundamental reason for conducting mobile network traffic analysis and prediction in subsequent network optimisation.

In recent years, cellular communication technology has experienced rapid development. Both network capacity and network data volume have grown significantly. In terms of network capacity, the global number of mobile devices and cellular network connections is projected to rise by nearly 60% from 2018 to 2023, reaching 13.1 billion devices [63]. Moreover, 10% of these devices are expected to be 5G-enabled [63], offering ultra-high transmission rates and ultra-low latency. Such a tremendous number of connected devices has contributed to the explosive growth of wireless network traffic. According to a whitepaper published by Cisco, it is estimated that by 2022, the total monthly mobile data traffic will increase to 77EB, which is seven times the growth compared to 2017 [62].

Despite the efforts made by mobile network operators to meet the ever-increasing network capacity and transmission bandwidth and ensure QoS through extensive deployment of small cells and technological innovations in communication terminals and channel coding [23], it remains challenging to keep pace with the explosive expansion of network data demands. Consequently, optimising, managing, and allocating network resources at the software level has become a feasible solution [24].

Currently, mobile network traffic prediction primarily focuses on two aspects: increasing network capacity and reducing OPEX. These aspects include network performance optimisation and base station power consumption control. Network performance optimisation involves preemptively estimating and allocating resources. For example, dynamic resource allocation [24] and wireless radio spectrum allocation [24]. Furthermore, reliable predictions of network load and load distribution are necessary for adjusting base station power or establishing Unmanned Aerial Vehicle (UAV) base stations when the network load becomes excessive [25]. In 5G networks, network slicing also relies on traffic prediction for flexible allocation and configuration of virtualised networks [26]. The deployment and resource management of Multi-Access Edge Computing (MEC) is similar, utilising predictive data to select functions for local network offloading [27]. Additionally, network traffic prediction plays a decisive role in self-configuring and self-optimising systems within SON [68]. On the other hand, power consumption control of base stations primarily assists mobile network operators in reducing OPEX [28]. One of the most representative approaches is base station sleep strategies. By predicting the inactive trends of network traffic, base stations can be powered off or reduced power to achieve energy saving [28]. Conversely, base station admission control utilises predictions of network traffic peak hours to allocate access among base stations, ensuring energy balance across the network under heavy loads [28]. It is evident that mobile network traffic prediction plays a crucial role in supervision, management, and allocation within the latest networks. Its performance directly impacts the overall system operation and is particularly significant.

2.1.1 The Features in Mobile Network Traffic

Due to the significant role of mobile network prediction in network performance and OPEX control, the characteristics of network traffic have been extensively studied. So far, the features of network traffic used in prediction models can be primarily classified into two types: temporal and spatial. When describing temporal characteristics, the focus is mainly on periodicity. The periodicity of mobile network traffic is initially discovered by observing a large number of base station traffic logs [10]. Generally, it shows higher traffic consumption during the daytime and lower consumption at night. Wang *et al.*. [17], through autocorrelation analysis, demonstrated the regularity of human activity patterns in wireless network usage. Additionally, numerous studies have analysed and validated the periodic load of traffic at the base station aggregation level [3] [8] [9] [11]. It is worth noting that while network traffic has been recognised as periodicity, the specific scale of the periods still requires detailed investigation. Xu *et al.* [29] conducted separate traffic analyses for each day of the week and discovered differences between weekends and weekdays. Moreover, special periods such as holidays have also been found to affect traffic trends [30].

On the other hand, the spatial characteristics of mobile network traffic have been gradually explored as research has progressed. Wang *et al.* [31] focuses on analysing the traffic similarity between base stations and utilises the characteristic of high similarity in proximity to optimise prediction accuracy. Taking a more macroscopic perspective, Zhang *et al.* [32] discovers the tidal nature of mobile network traffic within cities. As time progresses throughout the day, a certain area's network traffic experiences a synchronised increase or decrease, indicating a tidal phenomenon. It indirectly verifies the relationship between distance and traffic similarity between base stations. Additionally, different regions exhibit distinct traffic patterns. Zhou *et al.* [32] conducte traffic monitoring in various regions of urban and indicated that the characteristics of different regions affect traffic trends. Furthermore, Qiao *et al.* [18] illustrated that the proportion of traffic generated by different applications varies across different regions through the exploration of application-level traffic data.

Overall, the aforementioned characteristics of mobile network traffic have been proven to be crucial for the accuracy of predictions. It is through the discovery of these features that the prediction of such traffic data becomes possible. Additionally, it should be noticed the potential influence on feature bias and correlations by sampling from various scales of network traffic.

2.1.2 The Traffic Data Generated in Mobile Network

Due to the large scale and complex architecture of mobile networks, the traffic data generated from different sides of the network exhibit distinct characteristics. In general, there are two kinds of data generated in mobile network: traffic data and extra data. Traffic data is collected based on different ports and can be classified into network-level, cell-level, and application-level aspects. Network-level aggregated traffic refers to the traffic at the core network level [5]. Such data characterised by significant periodicity, can be used for traffic prediction through statistic-based modelling [22]. Regarding mobile network traffic, the focus is primarily on cell-level and application-level data obtained from the RAN side. At the cell level, mobile traffic data is collected based on temporal and spatial aspects. Temporal data collection typically involves sampling within a certain time frame with uniform time intervals [22]. The length of the time intervals depends on the overall historical data span and is usually set at hours [22]. In terms of spatial aspects, the data also needs temporal information and is annotated with geographic location stamps. The geographic location stamps are derived through classification or clustering algorithms, as the geographic information is highly discrete and not suitable for predicting traffic at the base stations or grids in the mobile network, which would increase computational complexity [34]. Application-level traffic is collected from the base station side as well. Unlike aggregated mobile network traffic at the cell-level, application-level traffic is segmented based on the types of data applications. This segmentation allows for the presentation of not only spatiotemporal characteristics but also information about traffic composition and application-specific demands [9]. Qiao *et al.* [18] combined application usage patterns with spatial information to discover the fundamental reasons behind variations in data traffic within a region, providing a better explanation for local base station traffic loads. On the other hand, Li *et al.* [18] analysed application traffic characteristics to address region classification and other related issues.

In addition to considering traffic volume datasets, heterogeneous datasets have also been used for research purposes to gain insights into mobile network traffic analysis. One of the most important aspects is obtaining UE's geographic location information, which provides real-time user information and is combined with traffic analysis to address issues related to wireless network access [36]. Sun *et al.* [37] utilisd users' path information to assist in user behaviour classification to apply further analysis of mobile traffic patterns. Another type of data that can be used in mobile traffic prediction is online data, which is public and obtained through APIs. It includes geographic tag data, user activity data, event alerts, and so on [3]. For example, Yang *et al.* [64] demonstrated a positive correlation between social media data (Twitter), network traffic, and user numbers, which helps mobile operators obtain additional insights to predict network traffic.

In summary, the type of data in analysing directly influences the direction and granularity of traffic prediction. Currently, there is a growing emphasis on cell-level and application-level data, in line with the trend of network refinement. However, it has been found that working with such traffic is more complex and challenging, as evidenced by former research. Additionally, the potential correlations between user and base station location information and traffic characteristics are worth considering and exploring.

2.1.3 The User Mobility Analysis in Mobile Networks

From existing research, it can be observed that user mobility has a potential impact on the analysis and prediction of mobile networks. For example, it has focused on user mobility behaviour classification [38] and the analysis of data patterns during group movements. Although these researches approach user mobility from different perspectives, they both validate the auxiliary role of mobility analysis in mobile network traffic prediction. From the user's perspective, being able to predict the next or multiple steps of entering specific cells can better forecast the traffic load within those cells [37]. From the group perspective, as users disperse or gather in a specific area, the traffic also exhibits significant variations [39]. Clearly, both approaches require the delineation of regions and the identification of hotspot areas that hold predictive value.

It is necessary to note that the delineation of hotspot areas for user access in mobile networks should be distinguished from geographical location-based divisions [39]. Due to the complex nature of urban planning, it is challenging to directly classify user mobility characteristics based on urban region attributes [40]. Therefore, clustering-based algorithms on users' historical visit locations have become a viable approach for delineating regions related to user mobility, which is the user point of interest approach [34]. Periodic POI-related patterns are of great help for understanding regular network traffic monitoring, BS planning[**?**], caching strategy[88] and forecasting[89][90]. It is usually seen as important cross-domain data in scenario classification, which is to help the system modelling features.

The discrete geographical location data in the same destination and the fuzzy of the user's location attributes is a constant concern. Whether positioning through GPS or a wireless network, the geographical location data in the same place is likely to be inconsistent. Different from handling outliers caused by positioning drift, discrete geographic location data cannot be averaged by using context information. The widely applied method is to unify the location data of the destination by establishing a reference spot. The DBSCAN-based clustering method[103]and the kernel function-based clustering method[104] can determine the reference spot according to the spatial information and obtain appreciate results. However, the problem of users' location attributes is caused by similar user destinations but periodic behaviour differently. For example, the location of a gym is close to the residential area, leading to the fusion of a user's behaviours of different periods in the observation window. Li *et al.*[105] performed periodic detection through each spot to distinguish different periods. Periodic behaviour is capable of portraying a user's different attributes in the same reference spot.

Once the POIs are determined, PPM is applied to analyse user mobility behaviour. PPM typically identifies repetitive patterns in the single object's trajectory rather than used for multiple trajectories. Once located periodic behaviour of the object, it can be used to figure out the relevance of places[97], forecast future movements[85], anomaly detection[98], and compress trajectory data[95]. It is important to note that, due to the nature of POI clustering, locations that are visited only once will be overlooked [41]. While this is acceptable, as detailed trajectory prediction for every user is impractical due to computational complexity and network performance limitations [35], it is necessary to extract the most important aspects from users' mobility history data, particularly the information with significant periodicity. This information can be utilised to predict future periodic traffic in mobile network prediction. Thus, PPM becomes an essential method for mobility prediction [42].

Existing studies have categorised PPM into two key types based on the resource data being employed, namely: sequence data and spatiotemporal dimensional data. Sequence data, or called one-dimensional data, can be defined as an ordered list of elements. The attribute of the order is decided by its application domain, such as by temporal order or spatial distribution order. For PPM in sequence data, the max-sub pattern hit set [91] is a feasible method by creating a max sub-pattern tree to mine periodic patterns by two scans of the time domain sequences. Furthermore, period discovery plays a crucial role in the PPM of one-dimensional data. Periodic patterns will be more accurate when the more precise period is detected. Fast-Fourier Transformation and Autocorrelation are the two key types of period detection [92]. Automatic period detection identifies as many periods as possible, although this might cause false and redundant periods to emerge. For the spatial domain sequence that includes spatial distributions, such as spatial cellular network traffic loads pattern, POI check-in distribution, etc., the researchers choose to model the inner logistic correlation of geometric characteristics of an item to compensate for the ignored information in the time domain PPM algorithms. Uday Kiran *et al.* [93] proposes a flexible model that can pick the desired partial periodic spatial pattern with three constraints, and then optimises the computation cost of the proposed model by using the prior knowledge regarding the neighbours of objectives generated by these spatial patterns.

The spatiotemporal trajectory is represented as the format of location and time, which is comprised of three-dimensional data (l_x, l_y, l_t) where (l_x, L_y) represents the geographic location while l_t represents the visiting timestamp. The set of each three-dimensional element along time order is a spatiotemporal trajectory dataset. Data integrity is another key limitation of PPM, which includes the perspectives of time interval, and data sampling rate. Zhang *et al.* [100] applied Lomb–Scargle periodogram in binary sequences to handle irregular time intervals of the dataset. Other similar issues with PPM historical data include coarse GPS or wireless network location resolutions[101] and errors in geolocation and positioning devices which can generate noisy data sets or missing data[102]. The former analogises the process of wireless network localisation into propagation in a parallel Gaussian noisy channel to derive the corresponding localisation error bound and the latter proposes a learning algorithm to predict the missing points based on the Automatic Identification System (AIS) data.

In conclusion, user mobility prediction can provide benefits for mobile network traffic prediction. Algorithms based on PPM have been widely applied as feasible solutions for detecting periodic mobility. Currently, there are several factors influencing the performance of PPM, with the processing of location information being one of the key factors. Therefore, the challenge lies in how to annotate user location information, such as using POIs, to better complement PPM in order to discover periodic patterns more effectively for mobility prediction.

2.2 Machine learning in Mobile Network Traffic Prediction

2.2.1 Supervised and Unsupervised learning

In supervised learning, a training dataset is used to provide both input and output information, which refers to features and labels, respectively. Typically, labels are assigned manually, to construct the relationships with the input data. Therefore, supervised learning aims to model the object information that can best approximate the relationship between input and output observable in the data based on the training sample dataset. According to such input-output relationship analysis, it can help to establish a feasible model. Subsequently, the new input dataset will be fed into the proposed model to make predictions.

The regression and classification problems are two main fields of supervised learning. The former is to deal with continuous data while the latter maps the inputs to discrete labels. Logistic regression, naive Bayes, support vector machine (SVM), artificial neural networks, and random forests are the widely applied algorithms in supervised learning. In both regression and classification, the goal of these algorithms is to model the features or structure in the given dataset effectively and accurately. The performance of the optimal model will be evaluated under a specific standard. Once it generated appreciated results through training and testing datasets, it can be treated as an optimal model to map the input and its labels and to solve the corresponding issues.

Bayes' Theory: Bayes' theory plays a key role in the statistical analysis used to calculate conditional probabilities. The following function reveals the probability that a hypothesis h will be affected based on the new evidence n.

$$P(h|n) = \frac{P(n|h)P(h)}{P(n)} \tag{2.1}$$

where $P(h|n)$ represents the probability that a hypothesis h will be true in light of the new evidence n. It is referred to as posterior probability. $P(n|h)$ is the probability that the new evidence n will impact the hypothesis h, and $P(n|h)$ represents the probability before considering the new evidence, which is called prior probability. Lastly, $P(n)$ refers to the probability of evidence n [110].

Through Bayes' theory, a new understanding of probability has been revealed. It is thus frequently employed in many areas of study. Mobile network traffic prediction is a specific scenario which has been widely applied. Akoush *et al.*[111] is a pioneer in this field by combining Bayes' theory with Neural Network (NN) to create an enhanced learning process by generating parallel outputs and predicting the location of mobile device users. Compared with computing the probability of multi-outputs, Zhang *et al.* [**?**] directly introduced the Bayes' theory in the parameters evaluation of the forecasting model, which can further improve the efficiency and accuracy of the training process.

Another popular research direction for Bayes' theory is for classification purposes which is called Bayes' classifiers. Moreover, it is often assumed that the inputs are all independent of each other which helps to create the Naïve Bayes' classifiers. Bayes' classifiers have been used in recent times in the field of anomaly component/cell detection [112][113]. In addition, naive Bayes classifiers are widely used to detect content-aware information, such as view counts, and stable and popular User Generated Content (UGC) patterns [114] [115].

With regard to unsupervised learning, data without using explicitly-provided labels are learnt to model the inherent structure. As a result, the model needs to generate the form and value of outputs simultaneously according to the internal relationship between training data and its external conditions. Therefore, unsupervised learning is useful in exploratory analysis because it can automatically identify features in the dataset[116]. Due to the unlabelled training dataset, the samples of the dataset must be classified based on similarities depending on the measurement methods, such as Euclidean distance, Dynamic Time Warp (DTW), Pearson coefficient, etc. An optimal system needs to ensure that the similarity between data in the same class is as high as possible while that of data in different classes is vice versa. Thus it can provide initial insights for testing individual hypotheses in further studies. For practical purposes, there are many situations that it is impossible to know the label or categories of the sample in advance. On the other hand, there is no way to compare model performance but to evaluate the suitability of analysing scenarios. Clustering algorithms, density estimation algorithms and representation algorithms are the main directions of unsupervised learning algorithms[117].

Clustering can be considered the most important unsupervised learning problem, which involves arranging unlabelled data into groups by measuring similarities between data samples[116]. The potential patterns can be modelled through clustering algorithms.

K-means: The K-means clustering algorithm is widely applied to identify clusters by determining the centre point in an unlabelled dataset. It is a typical example of a distance-based clustering algorithm, in which the similarity evaluation index is calculated based on the distance between two samples. The equation below can be used to express the K-means algorithm:

$$min\sum_{i=1}^{K}\sum_{x\in(C_i)}||X_i-u_i||^2 \tag{2.2}$$

$$u_i=\frac{1}{|C_i|}\sum_{x\in(C_i)}x \tag{2.3}$$

In this equation, u_i presents the mean vector of a cluster C_i. Additionally, i represents the order number of clusters. x represents the points in a cluster Ci. K represents the total number of chosen cluster's centre points.

The K-means technique has been widely used with various algorithms[118]. Sun *et al.*[119] firstly proposes a two-stage method with the combination between K-means and a Self-organising map, which can identify seven patterns and cluster them to further analyse the fine-grained patterns in cellular network service. Zahra *et al.*[120] replaces the distance measurement with Pearson correlation in UGC clustering to increase the efficiency of multidimensional data similarity measurement. Nowadays, K-means remains a popular clustering algorithm for wireless network traffic prediction[121].

On the other side, K-means also has several issues worth exploring. The mean vector of clusters in K-means is obtained by using the heuristic algorithm. Therefore, the selection of the initial centroid location will influence greatly the final clustering result and computation time, which requires an appropriate location selection in the initial point chosen stage. If the selection is only random, it may lead to slow convergence of the algorithm. K-means++ algorithm is the optimisation method of K-means on the initialisation of the centroid. Depending on the intuition, the farther away the clustering centres are from each other, the better performance of clustering will have. Therefore, the next added cluster initial points are as far away from the trained mean vector locations as possible to improve the training efficiency. Wang *et al.* [122] applied K-means++ in taxi passenger hot spot mining problem in this method and finds Its superior performance compared to the traditional K-means.

Density-Based Spatial Clustering Algorithm with Noise (DBSCAN) is in terms of density-based spatial clustering of applications with noise, which is another widely used clustering algorithm till now [123]. The density of the database represents the number of points per unit area. Any area with insufficient density will be considered noise and thus removed while high-density region will be clustered. It categorises the data points in the region into core points and boundary points. Two hyper-parameters need to be defined in advance, namely neighbourhood radius ε and the Minimum number of points (MinPts). If there are at least

MinPts data points in the ε range of a sample data, this point is regarded as the core point, otherwise, as the boundary point. If there are other core points in the ε range of a core point, they will be clustered. The adjustment of such two parameters will affect the size and number of clusters.

The key difference between K-means and DBSCAN is that the clustering number of the latter method k does not need to be defined in advance. Moreover, the DBSCAN algorithm can detect abnormalities during the dynamic clustering [124], which is more robust in a practical environment. It's worth noting that considering in predictive power of a single traffic clustering and minimising the number of clusters in network traffic analysis, DBSCAN performs better than K-means clustering [125]. Nonetheless, there are some limitations of this algorithm [126]. Firstly, the non-homogeneous density of data points will affect clustering performance. Additionally, due to the noisy data handling capability, DBSCAN requires more processing time compared to K-means clustering especially in analysing large datasets. Finally, the same as K-means, two hyper-parameters need to be adjusted in model optimisation. Aftab *et al.* [127] takes these issues into account, and then proposes a hybrid framework with a combination of K-means and DBSCAN to improve the efficiency of clustering. Therefore, these two algorithms need to be employed according to data structure and applying scenarios.

2.2.2 Machine Learning-based Methods in Cellular Network Traffic Prediction

Wireless network traffic prediction, due to the nature of its periodic pattern and the OPEX associated with mobile operators, has attracted a lot of interest from both academia and industry. The more and more studies have adopted machine learning technologies for mobile network traffic prediction. Since the first paper on machine learning [128], it has been studied for over fifty years with many methodologies proposed for various analysis cases. Unlike the statistic-driven methods, machine learning-based methods package the mathematics models used for mobile network traffic prediction into opaque or semi-opaque black boxes, which need to be trained using historical traffic records [134]. Existing machine learning-based prediction methods consist chiefly of shallow learning methods and deep learning methods.

In the early stage, Local linear regression (LR) was used in traffic prediction [135]. Since LR is essentially a linear model, traffic mutations and its heterogeneous non-stationary still cannot be effectively modelled. SVR uses the inner product kernel function to perform a non-linear mapping from the input space to the feature space [136]. It realises the capture of the non-linear relationship to achieve the purpose of prediction. Nevertheless, due to the limited capability of feature extraction, SVR usually has to rely on the prior knowledge of the hyper-parameter depending on the input features, therefore these shallow learning methods cannot cope with complex practical prediction scenarios. Therefore, deep learning tools, through nested functions to achieve appropriate non-linear mapping, have also been leveraged for wireless network traffic prediction. Among them, the recurrent neural network (RNN), designed to convey information from one time step to another, is widely utilized in the analysis of nonlinear time series, such as natural language processing (NLP), audio and video processing. Nipun *et al.* [137] verified the dominance of RNN architectures in time series forecasting by comparing the performance of other algorithms through real mobile network traffic databases. Moreover, it demonstrated that long short-term memory (LSTM), an RNN architecture that alleviated the vanishing and exploding gradient issue by adding memory units, can further raise the accuracy of network traffic prediction. One of the state-of-arts methods is considering temporal and spatial dependence. Exploring the cellular network traffic correlation between the target cell and its adjacent cells, Zhang *et al.* [138] proposed a densely connected convolutional neural network (CNN) based model by treating traffic prediction as an image analysis problem. Similarly, by incorporating convolution operations into LSTM, Zhang *et al.* [139] applies convolutional LSTM (ConvLSTM) to extract spatiotemporal features. Due to the retention of advantages in the time series analysis from LSTM, it has an enhanced performance compared to traditional CNN. Complex and functional algorithms and increased cells did achieve the goal of system improvement. However, due to the existence of the *degradation* problem in the deep network [140], the system performance reached a bottleneck. The proper networks in the above articles are kept at a small number of layers, which also verifies this.

To solve the aforementioned problem, many researchers have focused on the framework optimisation of the deep learning-based model. The first kind of attempt is to integrate various learners to retain their strengths in capturing patterns. Madan *et al.* [141] utilises the AutoRegressive Integrated Moving Average (ARIMA) and RNN to process linear and nonlinear components in traffic, respectively. Then forecasts are merged to get the final prediction result. The second kind of attempt is to unite similar traffic patterns from other cells. Zhang *et al.* [142] introduces a multi-task learning model to share learned features of similar patterns. Feng *et al.* [143] designs feature extractors to obtain features from other similar pattern cells and then applies them to the LSTM model to capture the temporal information. The third one is to extract more potential dependencies by introducing external information. Currently, widely used external information are POI, social activity, and other communication services data (e.g. calling, SMS) [143] [144]

However, most of the existing deep learning-based methods are proposed for the aggregated traffic generated by all the applications, whose variations are relatively stable and regular. Besides the analysis of aggregated mobile network traffic, some application-level traffic or microcell traffic depict different patterns with obvious randomness temporally and different correlations spatially, which makes conventional methods become less effective or less reliable for such traffic prediction. It is ignored in most of the existing methods. Therefore, it is worthwhile to explore the prediction models for high-granularity mobile network traffic data. In the pursuit of optimising prediction performance, researchers have started exploring additional features and information. However, the specific information that needs to be extracted is still subject to further validation.

2.2.3 Multi-task Learning Framework

As a data training method, MTL focuses on developing machine learning models in which shared representations are used to train data from many different sub-tasks simultaneously. With modelling the shared and independent patterns between these sub-tasks, it can ultimately raise the system efficiency and even the performance, which addresses the common weaknesses of other deep-learning frameworks such as modelling multi-dimensional datasets, computation cost and expanding the pattern extraction. The human learning process is applied in the concept of the MTL framework, which helps to incorporate information from various domains[145]. Meanwhile, there are challenges associated with the performance of multiple tasks according to the structure. Sub-task correlation is one of the most pressing issues in MTL. Low-correlation sub-tasks may conflict in feature extraction, which leads to degrade the performance of the system[146]. It is worth noting that in the case of MTL, two high-correlated sub-tasks are low-correlated to other sub-tasks. Thus how to reduce the negative impact on the overall system and how to measure the low correlation sub-tasks are the questions that need to be considered.

MTL has some properties suitable for sequence-related prediction. From the perspective of spatiotemporal correlation, Qiu *et al.* [158] and Huang *et al.* [159] both have verified that by performing a homogeneous MTL mechanism, using traffic of each cell as sub-tasks, the result of extracting spatial features can be achieved. It is even better than the results of convolutional networks, which validate the possibility of using multiple single-feature tasks to achieve multi-feature mining. In addition, Zhang *et al.* [142] proved that the similarity of different traffic will affect the performance of MTL in sequence prediction, and the method for traffic similarity measurement can effectively maintain the performance by increasing the sub-tasks. Furthermore, Zhang *et al.* [160] shows the efficiency of multi-position fusion and splitting training to transform features in MTL.

Benefit to the unique sharing parameter framework of MTL, it can be utilized in mobile network traffic prediction by introducing appropriate sub-models for sequence datasets. MTL leverages the shared parameter learning of correlated features across multiple tasks to optimise predictions. It not only increases learning efficiency and accuracy but also reduces the size of the model through parameter sharing. However, current research lacks studies on how to determine the correlation between sub-tasks, the configuration of sub-tasks in multi-task learning, and the design of specific frameworks to improve prediction accuracy.

Chapter 3

Personal POI detection based on periodic pattern mining

This chapter presents a framework for predicting user mobility based on machine learning-based algorithms and user points of interest features. The framework includes extracting user POI with historical locations, utilising the periodic pattern Mining algorithm to capture periodicity features, and finally employing a sequence neural network algorithm LSTM for next location prediction. Through experimental results comparison, this chapter demonstrates that the framework improves the current research on predicting the next location for users. Additionally, it validates the significant potential of LSTM in sequence data prediction and illustrates the periodic characteristics of mobile device users.

3.1 Introduction

With the development of the internet and mobile communications industries, mobile handheld devices such as mobile phones are also rapidly being updated. The more and more sensors are integrated into these devices and improve users' QoS through these generated data. Among them, the next destination prediction is one of the most concerning issues. By incorporating such location predictions into different types of users' activities, the system can proactively provide instant services for upcoming activities, such as mobile traffic prediction, context-aware recommendation, communication optimisation for ultra-dense populations, smart city services, etc. The aforementioned services are directly or indirectly related to population mobility. Therefore, predicting the next destination is crucial for population mobility-related network services.

Exploring users' periodic mobility patterns is one of the important factors to improve the accuracy of the next destination prediction. Research has shown that a significant portion of users' visited locations are relatively fixed [43], which aligns with common knowledge in daily life. By identifying these locations and uncovering the visiting patterns associated with them, the performance of the next destination prediction can be expected. Therefore, feature extraction for the next destination prediction can be divided into two aspects: location extraction and PPM.

User POI can serve as a method for location selection. It is primarily considered due to the nature of mobile network services and the objective of next destination prediction. Although there are some issues similar to handover control that require more accurate geographic coordinates of UEs [44], when it comes to network traffic prediction and related network services, users are typically expected to stay in an area for a certain period of time to consume mobile network data. Therefore, the goal of next destination prediction is to identify locations where users spend a longer time and use a larger volume of data for prediction, rather than predicting user trajectories based on latitude and longitude information. Hence, in this thesis, POI extraction is considered for the purpose of identifying hotspots visited by users.

Currently, there are two widely used methods for extracting POI based on geographical information: statistical models such as Gaussian distribution extension models [162] and unsupervised machine learning algorithms like K-means and DBSCAN [45]. Among them, clustering algorithms are more popular because they allow for flexible parameter settings to determine the range of each POI.

In terms of POI extraction, since geographic location information is generated by mobile devices through GPS, cellular network base stations, and Wi-Fi Received Signal Strength Indication (RSSI), there are some fluctuations in the coordinates [46]. Even if users engage in periodic behaviours, there can be visible variations in their location due to the inherent uncertainties in geo-localisation. Therefore, it is necessary to further refine the clustering range for POI extraction. This paper proposes a statistical-clustering algorithm to optimise the specific division of POIs and eliminate noisy data for PPM.

Furthermore, most users exhibit repetitive patterns of appearing at specific times and locations [43], such as working in the office on weekdays or sleeping at night. Certainly, users may have multiple periodic behaviours during a time interval and the period of repeating the mobility pattern is various. Hence how to figure out the periodic pattern of such behaviours is the key element in location prediction research. Detecting multiple periodic patterns and addressing data incompleteness and random biases in periodic pattern data pose challenges. Inevitably, users also take some temporary actions, or the period of such actions exceeds the selected time range. It is hard to get the desired results of such behaviour predictions if only based on historical timestamps. To predict the stochastic behaviour, it typically requires high dimensional data, such as labelled POI or event detection to figure out the users' purposes, which is not the concern in this thesis

In existing research, there are two main approaches for using POIs in PPM: mathematical modelling and sequence prediction. Li *et al.* [162] utilisd probability distributions and Kullback-Leibler divergence to compute the associations between different POIs and estimate the likelihood of each POI within a certain time period. However, the generalisability of data patterns is not discussed. On the other hand, sequence prediction methods mainly focus on Markov Chains (MC). Rendle *et al.* [55], first addressed the next-basket recommendation directly by combining matrix factorisation with Markov chains, known as Factorising Personalised Markov Chain (FPMC). Cheng *et al.* [47] extended FPMC with embedded personalised MC and user mobility constraints to address issues such as user movement noise. Sequence neural networks, such as RNNs, have been incorporated into some researches to capture long-term information and provide comprehensive periodic analysis [48].

However, the aforementioned recommendation methods have their own limitations. Markov chain-based studies must address issues such as long-term periodic pattern discovery and feature updates. RNN models, due to their inherent drawbacks, are prone to problems like gradient vanishing and gradient exploding. Additionally, it should be noted that the depth of neural networks in these studies increases to achieve more accurate feature modelling, which poses challenges in terms of computational power and energy consumption.

Recently, the introduction of POI contextual information has further advanced the accuracy of prediction. Kong *et al.* [49] used spatiotemporal context and employs a Hierarchical LSTM model for the next POI prediction. The novelty of this paper mainly lies in its data utilisation. In the spatial context, it uses the user's visit session to fully record the historical visiting sequence. The importance of capturing temporal context is demonstrated by the different distributions of continued visits to areas after visiting target locations at different time intervals. Meanwhile, HST-LSTM uses transition matrices to describe dynamic time intervals and geographical distances. However, it should be acknowledged that these transition matrices cannot accurately represent the relationship between time and distance, as these are continuous data in the real world. Moreover, this research primarily focuses on establishing training data sequences while neglecting other data feature analyses. This presents some challenges for HST-LSTM to become the optimal solution for the next destination prediction. From these existing studies, it can be concluded that in addition to summarising the spatial features, the inherent temporal relationship of each POI should also be explored to supplement the weak temporal contextual information. This thesis investigates the temporal characteristics of each POI through the analysis of independent access data for reference regions, guiding the feature extraction of neural network models.

In addition, two fundamental assumptions based on the dataset are proposed to as follows.

- The time interval is fixed, and the location records are not interrupted.
- There is at least one periodic behaviour in the records.

The purpose of this article is to establish a personal POI for users by exploring periodic behaviours to optimise the performance of next-destination prediction. There are three steps in this algorithm to detect periodic behaviours and establish personal POI. The first step is to measure all of the possible periods in the reference region. The clustering algorithm and kernel method are applied to discover those reference regions. For each reference region, the movement data is transformed from a spatial sequence to an in/out binary sequence, which facilitates the detection of periods by filtering the unconcern location. Besides, according to our assumption, every period will be associated with at least one reference region. All the periodic behaviours can be found if I detect all the reference regions in turn. In the second step, we statistically model the periodic behaviour by the Bayesian function. Based on this model, underlying periodic behaviours are detected and reconstructed to reference the region probability list to establish the personal POI. Finally, LSTM is applied to predict the next destination using POI preference data, which serves as a validation of the effectiveness of the POI extraction method proposed in this chapter. The major contribution as follows:

- Proposing a suitable algorithm to establish a more delicate reference region by eliminating interference caused by location errors.
- Introducing an algorithm for independent binary analysis of each POI region to uncover the temporal characteristics of each POI and establish user POI preferences.
- Applying POI preference with contextual information into machine learning algorithms to improve prediction performance.
- Training by real-world data, demonstrating its effectiveness in improving the accuracy of next destination prediction compared to existing algorithms.

The rest of the paper is organized as follows. Section 3.1 introduces the outline of the framework, problem, and dataset. Section 3.2 explains the mobility pattern and constraints. Section 3.3 describes the method to discover the periodic behaviours (stage 2). Section 3.4 introduces the fundamentals of LSTM and applies it to the next destination prediction by training proposed POI preference. Section 3.5 shows the prediction results and section 3.6 concludes this framework

3.2 Proposed Model

PPM aims to help the system figure out whether the current position is UE's destination and how long it will stay by modelling spatiotemporal features of UE's historical movement.

Moreover, due to the characteristic of most people's daily lives, which usually spend a lot of time staying in a few regions for working and resting, the system needs to show strong confidence in the result of prediction in the long-term, multi-times and staying duration. Periodic mobility patterns, loosely defined as repeating activities at certain locations with some time interval [162], can be observed virtually by each person and achieve that demand. Once the period of visiting time is detected, the system can make predictions instantly, and the staying duration is possible to predict as well if we have enough timestamps. The cost is to regard non-frequent visiting regions as noise. It is acceptable due to the leak of such non-frequent visiting records, and the duration of staying in such areas is uncertain, and the destination can only be predicted with some specific conditions. On the other hand, other pattern mining models, which are destination recommendation by user's interest [163], new event destination by natural language from social media [164], next hour prediction [165], and specific condition prediction [166], cannot meet one or two of these requires

3.2.1 Problem statement

S_u is the set of records of user u, and each record $s_i \in S_u$. It is a 4-tuple $s_i = \{lat_i, lon_i, t_i, w_z\}$, where i represents the i-th record, lat_i, lon_i, t_i,w_z, represent latitude, longitude, time and day of the week, respectively.

After generating the records, this chapter aims to model user's periodic patterns through POI data. Hence, the reference region, a certain area that is frequently visited in the movement, is built as the user's POI. These reference region set is denoted as $R = \{y_1, y_2, \dots, y_n\}$, where n represents the number of reference region. Each y_n is location segment as $y_n = \{ylat_n, ylon_n, ylab_n\}$, which $ylat_n$ and $ylon_n$ is the center coordinates of the n-th reference region, and $ylab_n$ is the label of n-th region. A period C represents the possible periods in the reference regions. It can be described as $C = \{c_1^1, c_1^2, \dots, c_n^m\}$, where c_n^m is the m-th period time in the reference region r_n. The variability in the number of periodic patterns within each region is attributed to users exhibiting different behaviours in different regions.

O is the set of user periodic behaviour generated by the proposed PPM framework as, $O = \{\dots,(t_i,w_i,y_n,p_i^n),\dots\}$, where p_i^n represents the probability that the UE is at the reference spot $rlab_n$, at relative timestamp t_i.

Finally, by calculating the p_i^n for each specific time within each reference region, it can obtain the probability of the user visiting a certain region, thereby understanding the temporal characteristics. Therefore, the following three objectives need to be accomplished to explore users' periodic behaviour:

- The locations of region R need to be detected.
- The period set C needs to be measured.
- The user POI preference O needs to be calculated.

It can be known that personal periodic destination detection needs a huge amount of data to train the system. Four volunteers generated the trajectory data as personal mobility pattern through Google Map 'location history' function on Android operating system, and other sign in method as back up and propose a novel algorithm to measure three mentioned parameters. All data is divided along with hourly timestamps. If there is missing data, it will be the same as the previous state until the position is recorded again. Total data length has 31 days, the first 28 days dataset is constructed as training data, and last 3 days dataset is constructed as testing data.

3.2.2 Mobility data and constraints

In this model, s_i is collected through GPS, usually recording location information in units of time interval. The geolocation information primarily includes latitude, longitude, and time. The advantage of using GPS to collect position is that it directly provides latitude and longitude coordinates, and the recording time interval which can be adjusted, making it easy to capture user trajectories. However, there are three factors that may affect the data quality and need to be considered in advance: data missing, data drift, and uneven time intervals.

Data missing refers to the absence of some records in the sequence dataset. Data drift is defined as sudden significant deviations in latitude and longitude or random distribution within a certain area. These two issues are mainly caused by signal interference or problems with recording devices' computational capabilities [50]. Currently, such problems have been extensively researched, and mature algorithms for data processing exist. Since this paper involves POI extraction and is not highly sensitive to geographic location information, and the research topic does not focus on these issues, this chapter will directly use methods that have been applied in multiple studies [51] [52], namely mean imputation and velocity anomaly detection, to address the above-mentioned problems. The former involves using the average of preceding and succeeding sequence data to find the intermediate value. The latter calculates the velocity based on the distance and time of preceding and succeeding data, and if the velocity exceeds a threshold, it is considered an anomaly and will be modified.

Uneven time intervals can indeed affect the validity of modelling. For example, if the first five sequence data are collected within one hour, while the next five are collected over several days, it is evident that such sequences cannot provide useful information due to the inconsistency in temporal dependency. This issue may be due to device issues or recording strategies. Some devices, for energy-saving purposes, pause the recording of location information once the user is not moving [41].

Based on the characteristics mentioned above, this chapter applies the method of averaging coordinates. Specifically, within a unit time interval, the existing records are averaged to address the issue of uneven time intervals. The average coordinate during the time interval j can be defined as:

$$\overline{\text{lat}}_j = \frac{1}{K}\sum_{k=1}^{K} \text{lat}_j, \forall k \text{in time interval} j \tag{3.1}$$

$$\overline{\text{lon}}_j = \frac{1}{K}\sum_{k=1}^{K} \text{lon}_j, \forall k \text{in time interval} j \tag{3.2}$$

where K represents the amount of data in time interval j. If there are no records within a unit time interval, the calculation of $\overline{\text{lat}}_j$ and $\overline{\text{lon}}_j$ will be average the positions from the preceding and succeeding time intervals as follow:

$$\overline{\text{lat}}_j = \frac{1}{2}(\overline{\text{lat}}_{j-1} + \overline{\text{lat}}_{j+1}) \tag{3.3}$$

$$\overline{\text{lon}}_j = \frac{1}{2}(\overline{\text{lon}}_{j-1} + \overline{\text{lon}}_{j+1}) \tag{3.4}$$

Due to the focus of this chapter not being on data pre-processing, the subsequent sections describe the actual samples using the dataset that has been pre-processed using the aforementioned methods.

This chapter aims to detect periodic movement patterns and establish POI preference. To make the model clearer and more reasonable, there are some assumptions and constraints. First, following the settings of existing research [162][161], it assumes that each region has at least one independent visit cycle (if there are multiple cycles, the overall cycle will be the least common multiple of them). In order to detect periodic regions, the proposed model combines geographic and time information. This is necessary because estimating a period from mixed periodic information is not possible solely from the location information, especially when the regions are close to each other. For example, a gym may have three different classes scheduled once a week, once every three days, and once every eleven days, respectively. Such mixed and non-multiple periodicities may lead to dispersion and difficulty in extraction. Therefore, this chapter usually selects the most evident periodicity for exploration. Furthermore, if a region does not exhibit any visit period, it will not be considered as a reference region.

Secondly, this chapter focuses on daily periodic behaviours rather than monthly or yearly behaviours. It is well-known that the most common periodic behaviours occur at workplaces, restaurants, gyms, entertainment venues, and homes, with daily or weekly cycles[41]. Additionally, the cost of detecting periodic events, such as data collection, storage, and training, increases rapidly from daily to monthly, and increases even more significantly when detecting yearly events, as it requires much longer historical records. In this model, the historical records are not sufficient to study monthly and yearly periodic behaviours. Therefore, the standard period c_n^m used in this paper is one day (24 hours).

Finally, the periodicity duration needs to be shorter than a certain threshold compared to the length of historical data. Through multiple attempts, it has been found that with one month of historical data, periodic behaviours need to occur at least three times or more to be detected. Therefore, periodicities with occurrences below three times will not be given significant attention, unless the historical data time is extended.

3.2.3 Locating reference region

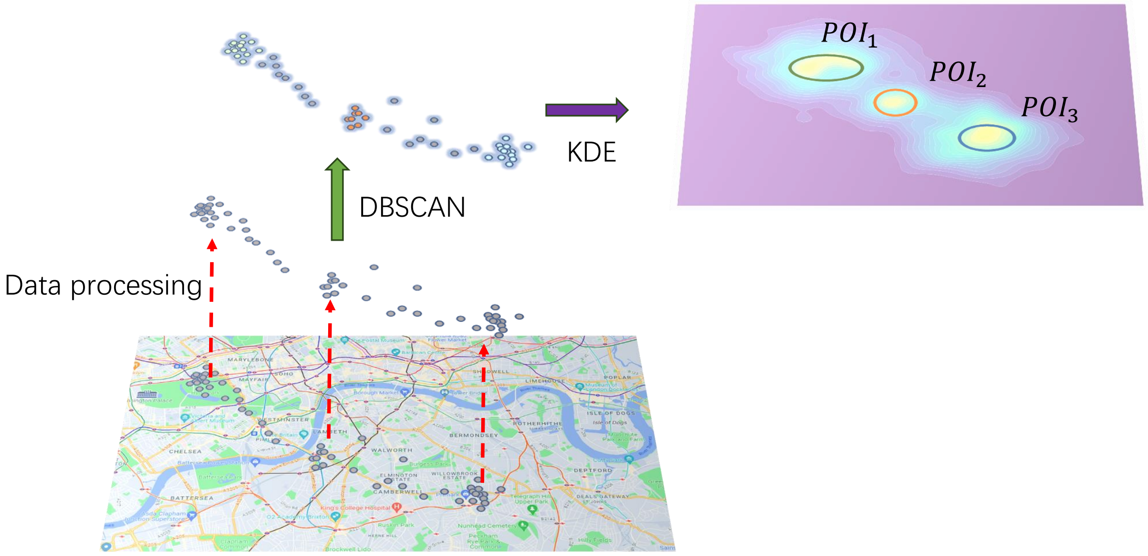


Fig. 3.1 User trajectory and POI extraction

Due to the periodic movement of objects, certain specific locations will be repeatedly visited, and spatial information is prioritized for extraction to form reference spots. Reference regions represent dense areas where users have more historical location records. A common statistical method to obtain reference regions is by calculating the point density for each region. However, this approach can be computationally expensive, and neighbouring regions may interfere with each other.

To address these challenges, this chapter introduces a more intuitive clustering algorithm that measures the distances between points as the criterion for clustering. This method initially identifies clusters of data points that are close to each other, while sparse points are considered as noise and removed. Since the data has been sampled with a unified time interval, sparse points are reasonably attributed to short-term stays or ongoing movement, which are less relevant for mobile network service-based mobility prediction.

In the clustering algorithm, some edge points within data clusters may be included due to special circumstances. Once independent regions are segmented, density calculation is performed for each region to determine the exact location of the visited area, further refining the reference regions.

DBSCAN and K-means are two widely used clustering methods. In this chapter, DBSCAN is introduced as the clustering method for two main reasons. Firstly, compared to the K-means method, DBSCAN does not require a predetermined number of clusters, which significantly reduces the computational complexity. In K-means, the number of clusters needs to be set as a hyperparameter or optimised using the Elbow method based on computing the Sum of Squared Errors (SSE), which involves multiple global computations to find the optimal solution. It becomes impractical when dealing with large datasets. On the other hand, DBSCAN does not have such a requirement, making it more efficient for this specific task. Secondly, DBSCAN's ability to determine the density of surrounding points helps exclude low-density points which are considered as noise. In contrast, K-means classifies all points without distinguishing noise points. In the context of this chapter, the path records in user movements are useless for periodicity detection and are treated as noise. By reducing the inclusion of noise points, DBSCAN allows for clearer extraction of periodic patterns, making it more suitable for the POI establishment purpose of this chapter.

DBSCAN uses two parameters to determine the clustering scope: the radius of the neighbourhood, denoted as ε, and the minimum number of points required to form the core point, denoted as MinPts. In DBSCAN, each point is checked within a radius of ε to count the number of points in its neighbourhood. If the count exceeds MinPts, the point is considered a core point and can absorb other points into the cluster through density-reachable and density-connected relationships. Points with a count lower than MinPts and not included in any cluster are marked as noise. Finally, by finding all the points connected to the core points, we obtain the clusters and noise in the dataset.

Indeed, it is important to note that adjusting the two hyperparameters, ε and MinPts, can significantly affect the number and scope of clusters. Setting these parameters too large or too small can reduce the clustering efficiency. Currently, the values of ε and MinPts are often determined empirically based on prior knowledge. Even so, DBSCAN's efficiency remains considerable compared to the global computation required by K-means.

Due to the influence of some edge points in DBSCAN, it is possible to include some neighbouring reference areas' noise points. Therefore, this chapter further proposes Kernel Density Estimation (KDE) to determine the centre and the range of reference region. KDE is a statistical method to estimate the probability density of random samples. Each sample is considered as generating from some distributions with an unknown density f and it accumulates a kernel density estimator at each sample. If the user has frequent activities at one place, this place will have a higher probability to be its periodic behaviour occurrence region. In this chapter, the normal distribution kernel function is used in KDE. Meanwhile, because the position of the user is in a two-dimensional space, it is considered as 2D KDE . Therefore, the probability density estimation value, denoted as $\hat{f}$, at coordinates lat_i and lon_i, is calculated as follows:

$$\hat{f}(\text{lat}_i, \text{lon}_i) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{\overline{\text{lat}} - \text{lat}_i}{h}, \frac{\overline{\text{lat}} - \text{lon}_i}{h}\right) \tag{3.5}$$

Where n represents the number of samples in the dataset. h is called the bandwidth, which controls the smoothing of the estimation. $K(a,b)$ is typically a normalised probability density function. The Gaussian kernel is applied here:

$$K(a,b) = \frac{1}{2\pi} e^{-(a^2+b^2)/2} \tag{3.6}$$

Fig.3.1 shows the process of the proposed method to determine the reference regions. Firstly, the historically recorded locations of the users are clustered using the DBSCAN method, which distinguishes different clusters and identifies noise points. After removing the noise points, for each cluster, the KDE is employed to estimate the probability density. This approach helps further reduce noise caused by model bias in the clustering model.

As depicted in the figure, the contour lines represent different probability density levels, indicating that the reference region can be determined by selecting the contour line with the top-$p\%$ density value. The larger the value of p, the larger the size of the reference region. To determine the optimal value for p, experimental comparisons are conducted in later sections. The results demonstrate that an efficiency peak is achieved when $p\%$ equals 60%.

3.2.4 Periodic behaviour detection

After detecting a set of reference regions, we further propose a method to obtain the potential periods within each region. To reduce complexity, in a single region, the user's mobility pattern now can be transformed as a binary sequence $B = b_1\ b_2 \ldots b_n$, where b_i = 1 when the user is within the reference region at ith timestamp and 0 otherwise.

As can be seen in [92], there is a popular method to detect periods in a discrete signal sequence which uses Fourier transform and autocorrelation. Fourier Transform (FT) is used to decompose a function from the time domain to the frequency domain. However, it suffers from a kind of low-resolution problem in the low-frequency region which is caused by the length of the period. 'Spectral leakage' is another issue caused by unsynchronised sampling frequency and signal frequency, which leads to false positive values in the periodogram. These problems can be solved by autocorrelation which can accurately estimate both short and long periods, but it is difficult to determine the essential period due to the issue of autocorrelation. Therefore, the combination of FT and autocorrelation is to examine whether the candidate period from the periodogram lies on a hill of the autocorrelation. If it does, we can consider it as a valid period otherwise it is a false alarm. Hence in this paper, we apply this method to detect the period of reference region visiting.

In Discrete Fourier Transform (DFT), we transform the sequence B to complex numbers $X_1, X_2, X_3, \ldots, X_n$, which is a set of sequence Fourier coefficient $X(f_{\frac{k}{n}})$. The periodogram P is provided by the squared length of each Fourier coefficient:

$$P(f_{\frac{k}{n}}) = \left\| X_{f_{\frac{k}{n}}} \right\|^2 \tag{3.7}$$

Where $P(f_{\frac{k}{n}})$ is the power of the frequency $\frac{k}{n}$. In this equation, the higher power it has, the larger probability the corresponding frequency is the period. Hence, it can be tested from the largest $P(f_{\frac{k}{n}})$, and we normally check top-four frequencies.

Given periodogram $P(f_{\frac{k}{n}})$, autocorrelation needs to be calculated to determine the exact period in the time domain, because a single value $X_{f_{\frac{k}{n}}}$ in the frequency domain corresponds to a range of periods $[\frac{n}{k} \cdots \frac{n}{k-1}]$ in the time domain.

Autocorrelation Function (ACF) is used to examine how the similarity between a sequence and its previous values for different τ lags:

$$\mathrm{ACF}(\tau) = \frac{\sum_{t=\tau+1}^{n}(b_t - \bar{b})(b_{t-\tau} - \bar{b})}{\sum_{t=1}^{n}(b_t - \bar{b})^2} \tag{3.8}$$

Where n is the length of the time series data, b_t is the value of the time series at time t, $b_{t-\tau}$ is the value of the time series lagged by k time points, and $\bar{b}$ is the mean of the time series data. The autocorrelation function (ACF) takes values between -1 and 1. When the autocorrelation coefficient is close to 1, it indicates a positive correlation, while close to -1 indicates a negative correlation. A value close to 0 suggests no linear correlation. The autocorrelation function is commonly used in time series analysis to assess the presence of periodicity or trends in the data. In this chapter, it is employed for assisting in the detection of periodic behaviour.

Therefore, when the periodogram provides a period range $[a,b)$, it can transform them to $[ACF(a), ACF(a+1) \cdots ACF(b-1))$ and test whether there is a peak. If there is, time is treated with a maximum ACF value as a valid period.

3.2.5 The probability of periodic behaviour and duration

Given the period in a specific region, we aim to find the duration of stay in each period. The method we used to solve this problem is the Bayesian approach which is a kind of machine learning algorithm that the parameters can be updated with a training database. The value generated by the Bayesian equation in this paper is to estimate the probability distribution

$p(q|y)$ over the probability of event occurrence q when the historical records show the event happened with y times.

Where the meaning of each factor is introduced in Chapter 2. When the probability distribution is measured, we can calculate the expected value of q under the posterior which is the probability of behaviour happening in the specific time interval. If there are several intervals with a high probability of q, we can predict the duration of the user visiting the region in one period.

In this paper, because it only detects the periodic visiting, it can be separated and analysed of the behaviour in the reference region instead of the behaviour along the timeline. Hence, due to the binary record of user visit, it can predict sequence B is a set of samples generated by a binomial distribution which is the likelihood $p(y|q)$in the Bayesian function. For prior distribution, it considers $p(q)$ as the beta distribution which can represent three scenarios of q. In this chapter, the probability in a binomial distribution is regarded as unknown for straightly showing the changes of the database. In Bayes' rule, if a likelihood (binomial) - prior (beta) pair is conjugate, the posterior has the same form as the prior without considering the form of margin likelihood $p(y)$ which is:

$$p(q|y) \propto \left[\binom{N}{y} q^{y}(1-q)^{N-y} \right] \times \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} q^{\alpha-1}(1-q)^{\beta-1} \right] \tag{3.9}$$

$$p(q|y) = \frac{\Gamma(\alpha+\beta+N)}{\Gamma(\alpha+y)\Gamma(\beta+N-y)} q^{\alpha+y-1}(1-q)^{\beta+N-y} \tag{3.10}$$

Where α and β are parameters used to control the shape of the resulting density function which means that both are equal to one in this paper to represent unknown prior. N is the total number of samples. The expected value of q under posterior is:

$$E_{p(q|y)}\{R\} = \frac{\alpha+y}{\alpha+\beta+n} \tag{3.11}$$

Then the system measures the probability of event occurrence in each time interval within the period. If there are two periodic behaviours in this region, we need to use Equation (3.7) to estimate the new probability p_i^n which is accumulated by two original clusters:

$$p^{n}{}_{i} = \frac{C_{n,1}}{C_{n,1}+C_{n,2}+\cdots} q_{n,1} + \frac{C_{n,2}}{C_{n,1}+C_{n,2}+\cdots} q_{n,2} + \cdots \tag{3.12}$$

Where $C_{n,d}$ is the width of samples in cluster n which is decided by a larger period. For example, if two periods are daily and weekly, the width of the daily cluster $(n,1)$ is seven days because the time of weekly periodic behaviour consists of each Monday to each Sunday. And the width of the weekly cluster $C_{n,2}$is the total number of weeks. And if the larger period is monthly, the width of the daily cluster is thirty days, and $C_{n,3}$represents the width of monthly in the same cluster n. From this equation, it is obvious that with the data increasing, the probability of larger periodic behaviour will domain the new probability because the smaller periodic behaviour is 'predicted' by the period detection algorithm which has less confidence than larger periodic behaviour. p_i^n is the expected probability for cluster n at i time slot.

Since there are three sub-tasks in the periodic behaviour mining problem, detecting the specific region, a period of user's behaviour pattern detection, and POI preference establishment. I propose the reference region extraction model shown in Algorithm 1, and the periodic behaviour pattern detection and POI preference sequence generate shown in Algorithm 2, in which each stage is used to target a sub-task.

Algorithms 1 and 2 show the general framework in this chapter. In the first stage, we try to find all the potential reference regions. Since a UE with periodic movement will repeatedly visit some places if we only consider the spatial information, the reference region will have a higher density of points than the other regions. Then, for each potential reference region, the periods are detected. Finally, for every period c_n^m, we further measure the probability of periodic behaviours occurring in each interval and then combine the time of high probability periodic behaviours to find the duration of it.

3.3 Research process and case study

In this section, we will apply these solutions to analyse the model simulated in the previous section and estimate the performance. Therefore, according to the outline of the simulation,

Algorithm 1: Periodic Behaviour Detection and Prediction

```
Input: A geo-location and timestamp sequence $S_u$, $\varepsilon$, Minpts
Output: Reference region sequence $R$
Initialisation;
$s_i \in S_u$;
Creating empty sets $R$ and $R'$ = [] ;
$n = 0$;
$n' = 0$;
Mark all the elements in $S_u$ as unvisited;
for $i <$ length of $S_u$, $i$ ++ do
    Find a set of points within the range with $s_i$ as the centre and radius $\varepsilon$, N;
    if the number of set N, $E_n \geq MinPts$ then
        Mark $s_i$ core point ;
    else
end
for $s_i$ marked as core point and unvisited do
    Mark $s_i$ as visited and label it as $n$;
    for Pick an unvisited point $s_j$ in $s_i$'s N do
        if $s_j$ is core point then
            Mark $s_j$ as visited and label it as $n$ ;
            Adding a set of points within the range with $s_j$ as the centre and radius $\varepsilon$,
              N' into N
        else
            Mark $s_j$ as visited and label it as $n$
        end
        Append all $s_i$ labelled as $n$ into $R'$ $n++$
    end
    for $s_i$ marked as unvisited do
        Mark $s_i$ as visited and label it as $n$;
    end
end
for $n'$<$n$ do
    Withdraw all the $s_i$ labelled as $n'$;
    Detect potential reference regions in sub-areas based on location stamp density
      using Kernel Density Estimator (KDE) with top-$p\%$;
    Calculate the centre coordinator of these $s_i$;
    Form a sub-set $y'_n = \{ylat'_n, ylon'_n, ylab'_n\}$ Append $y'_n$ into $R$;
    $n'$++
end
return $R$
```

Algorithm 2: Reference region extraction

```
Input: Reference region contextual sequence data $R$, the number of region $n'$
Output: POI preference sequence $O$
Initialisation;
# Stage 1: Periodic behaviour pattern detection;
h = 0 $B$ = [] for $n = 0, n < n'$ do
    Withdraw all $s_i \in R'$ have same label $n$;
    for $h$ < Historical data duration, $H$++) do
        if There is a visiting record at h in location n then
            Append 1 into $B$
        else
            Append 0 into $B$
        end
    end
    Find the period of user's behaviour in each interval and region during the entire
      modelling time with B;
    Record the period at $m$-th period time in the reference region $r_n$. $c_n^m$ ;
    Append $c_n^m$,into $R$
end
return $R$ and $B$
# Stage 2: Establish POI preference dataset;
$P$ = [];
for  $n = 0, n < n'$ do
    for $m' = 0, m' < m$ do
        Withdraw $c_n^m$ in $R$, which is the $m$-th period time in the region $n$;
        Withdraw all the period time slots in B in the region $n$, according to $c_n^m$;
        Use the Bayesian function to measure the probability $p_i^{n,m}$ of the user's
          position in each interval;
        Append $p_i^{n,m}$ into $P$
    end
    $n$++;
end
Merge the elements with the same $n$ and $i$ in P with Equation (3.13);
Append $P$ into $R$
return $R$ as $O$
```

we first set the parameters in the model to create a user's daily mobility trace in 28 days. Python is used to build this model, and the experiment is performed on a 3.0 GHz Intel i5 system with 8 GB memory. The system is Windows 10.

3.3.1 Dataset pre-processing

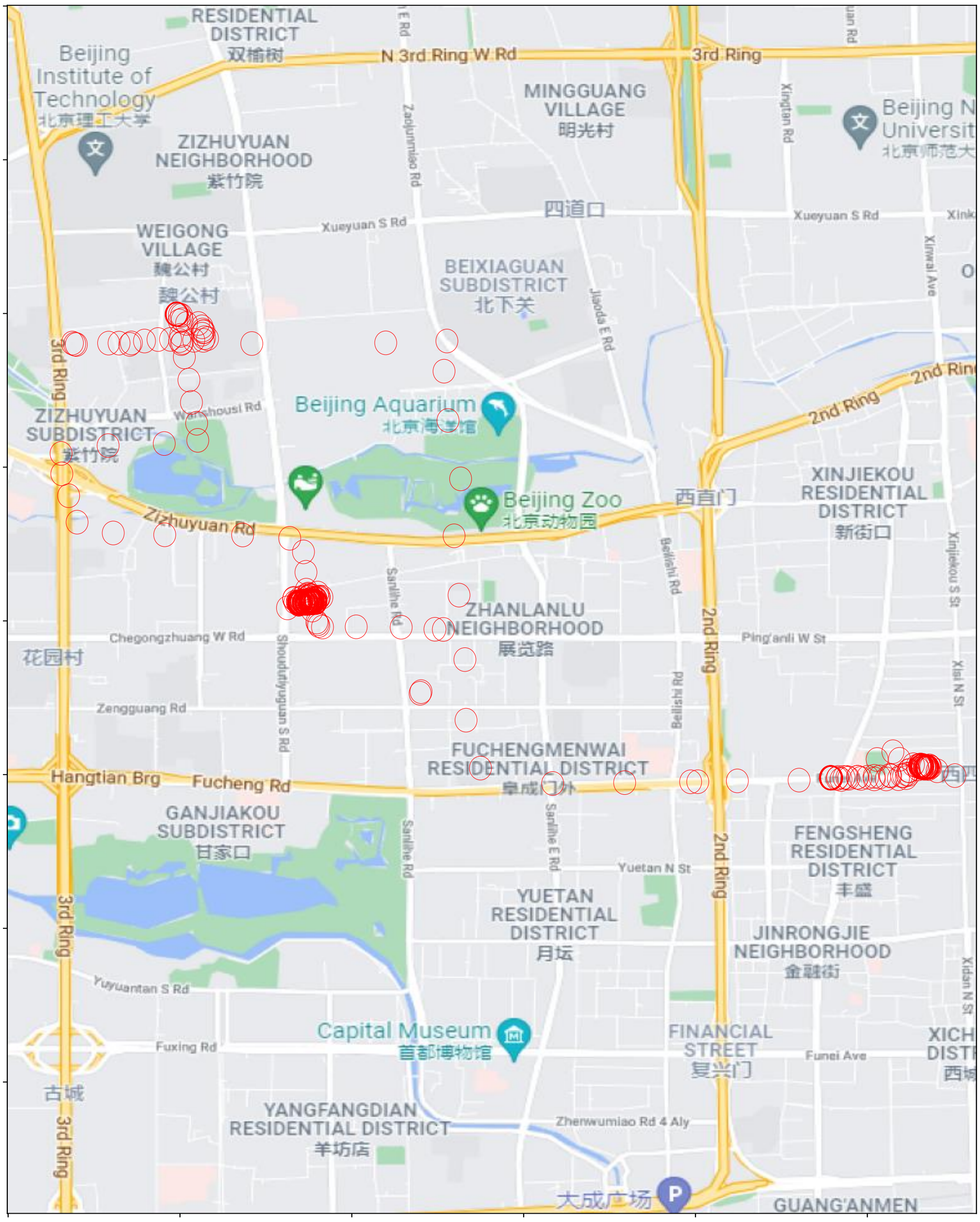


Fig. 3.2 A user's trajectory Geolife dataset

This thesis introduces a public dataset, called Geolife, collected by Microsoft Research Asia [53]. The project aims to explore urban user travel patterns and transportation, requiring location and time information. The dataset spans three years and includes observations from 182 users, recording their geographic location information at different time slots. The recorded data includes user IDs, GPS coordinates, and timestamps represented by numerical IDs, latitude, longitude, and time, respectively. Since this chapter mainly focuses on the analysis of the mobility patterns of individual users, it primarily retains the location and time data. Figure 3.9 displays the movement trajectory of a user for one day, with each point representing a recorded data entry. By overlaying this information on a map, the user's mobility can be visually observed.

As mentioned earlier, this dataset also suffers from some issues such as a small number of data missing, data drift, and irregular time intervals. To address the first two issues, this thesis employs mean imputation and velocity anomaly detection. For the irregular time intervals, it uses the mean coordinates to fill in the gaps and re-aggregates the data to record hourly intervals. Additionally, local time is converted into the day of the week.

Due to the incomplete trajectory records for some users, it is necessary to perform user selection. In the end, a suitable user's historical data is identified, consisting of 35 continuous days with hourly intervals, resulting in 840 data points. Each data point for a specific hour is represented as $s_i = \{\mathrm{lat}_i, \mathrm{lon}_i, t_i, w_i\}$, where $i \leq 744$. The variables lat_i and lon_i represent the latitude and longitude coordinates, t_i denotes the time information, and w_i represents the day of the week. The first 28 days dataset is constructed as training data, and the last 7 days dataset is constructed as testing data.

3.3.2 Reference region detection

As the previous section introduced, we need to use the density of points to detect the periodic behaviour regions.

First, as shown in Fig.3.3, DBSCAN is used for clustering. ε is set to 10, indicating a range of 10 meters, and MinPts is set to 5, requiring at least five neighbouring timestamps within the specified range. These parameter values are determined based on approximate

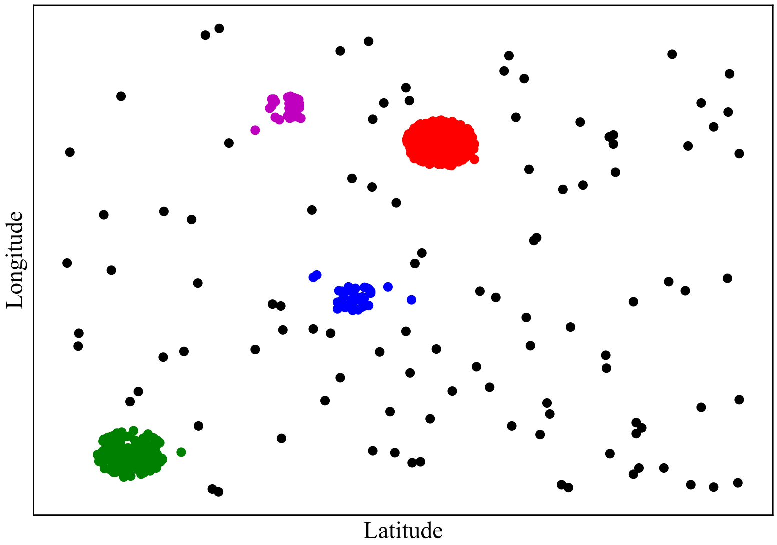


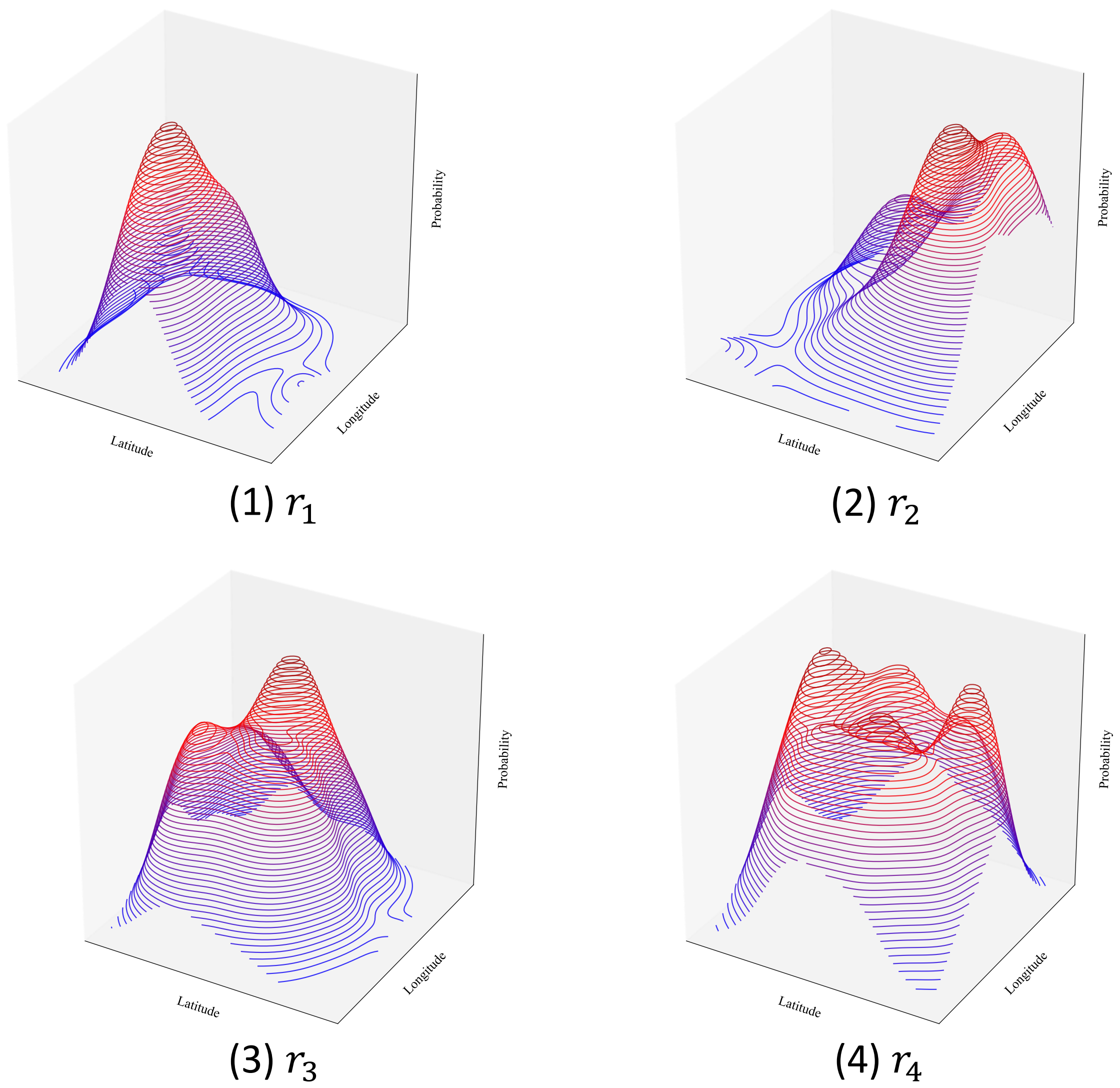
Fig. 3.3 The user historical trajectory clustered by DBSCAN

distance information and empirical knowledge. Four regions are identified as clusters, denoted as r_1, r_2, r_3, and r_4. By comparing with the actual map, it is observed that only r_1 corresponds to a residential area with a significant number of timestamps, suggesting it might be the user's home. However, the specific user behaviour cannot be determined solely based on this information; further analysis is required through the detection of periodic behaviours.

Next, KDE is applied to the DBSCAN clustering to locate the scale of the cluster range. Fig.3.4 shows the KDE distributions of the four regions, respectively, which visually demonstrates the expansion of the range with increasing top-$p\%$. To evaluate the performance of POI preferences, five different p values (20, 40, 60, 80, and 100) are selected for experimentation. The goal is to determine the optimal p value that yields the best results in terms of performance assessment.

3.3.3 Periodic pattern detection

This section explores the periodic behaviour of users in the four reference regions that were identified in the previous section. It first converts the data from each reference region into binary sequences representing in and out movements. Then, we use the FFT method to analyse the frequencies and detect periodic patterns. By combining the Periodogram and



(1) r_1 (2) r_2

(3) r_3 (4) r_4

Fig. 3.4 The KDE of four reference regions

autocorrelation algorithms, we can efficiently identify the periods in each region while avoiding spectral leakage. The results are significant, as we have successfully discovered the corresponding periodic patterns in each region.

r_1 stands out as a special case. It is located in a residential area, and the periodicity is observed daily, with most visits occurring during the night. It is highly likely that this region represents the user's home. Moreover, further periodicity exploration in r_1 did not yield additional discoveries.

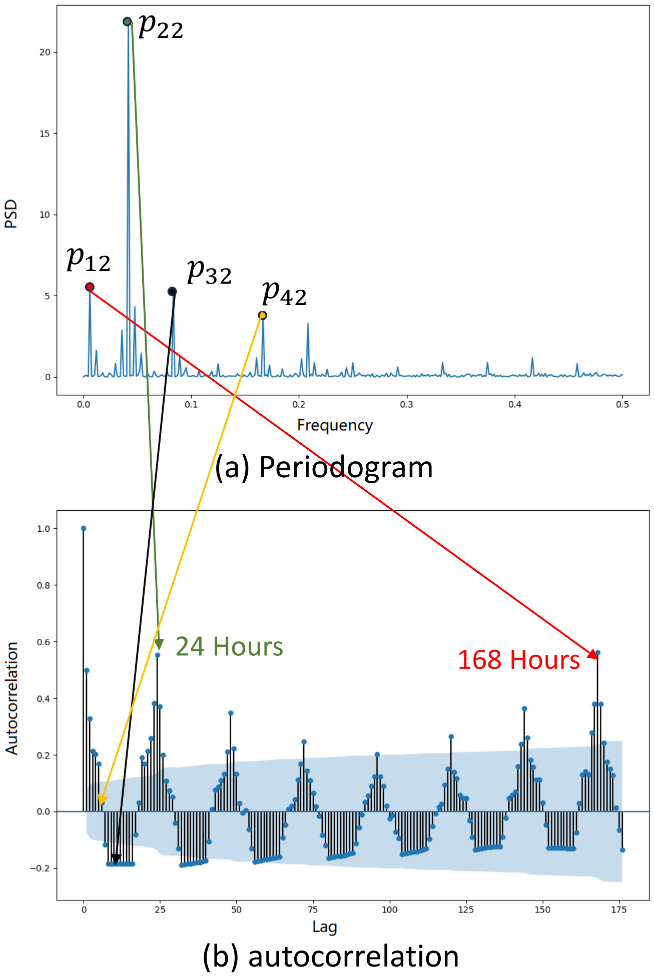
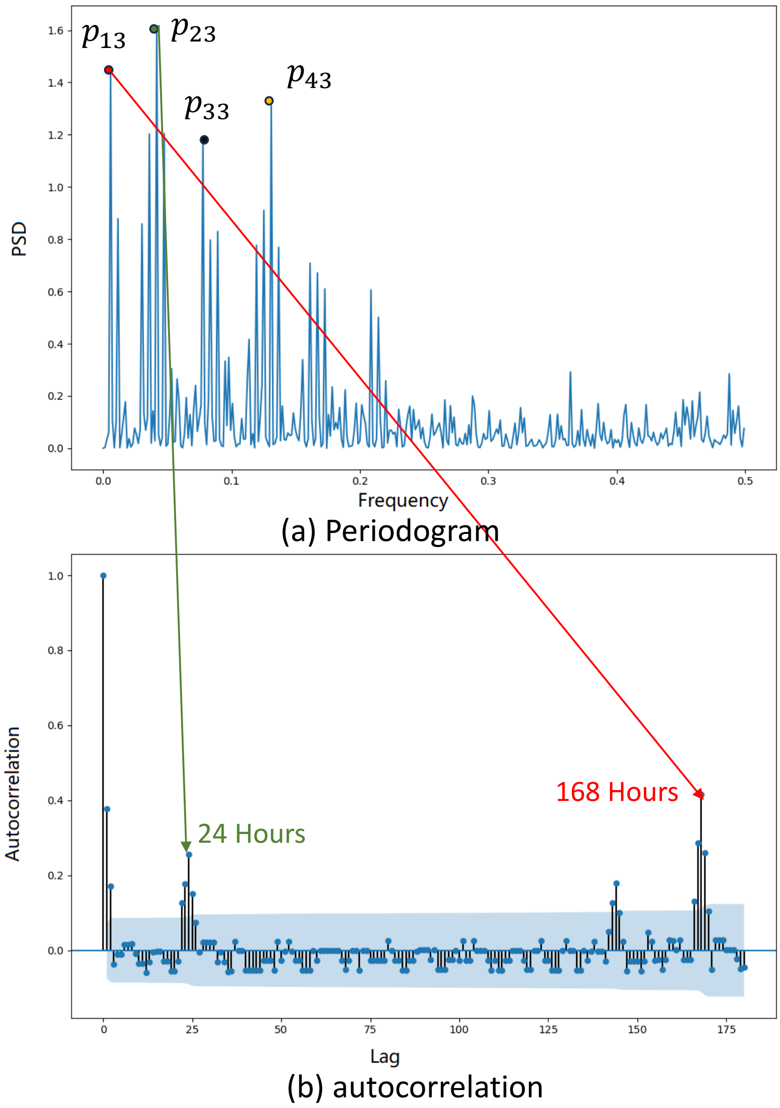


Fig. 3.5 Periodogram and autocorrelation to detect the period of user behaviour in reference region r_2, (a) periodogram of the pattern in r_2, (a) autocorrelation of the pattern in r_2

Fig.3.5 shows the periodogram and autocorrelation graph in r_2. In a, there are four frequencies with the top-four largest power spectral density (PSD) p_{12}, p_{22}, p_{32} and p_{42} which are the potential period of user's behaviour in r_2. However, in b, only p_{12} and p_{22} have peak points in the autocorrelation function which are 168 hours (a week) and 24.1 hours (a day) respectively. Hence, there are two periodic behaviour in r_2 which can be explained by that the user visits this region almost every day but disappears a couple of days regularly per week.

Fig.3.6 represents the period in r_3 which has 168 hours (a week) and 24 hours (a day) as well by comparing the periodogram and ACF. However, in the autocorrelation graph Fig.3.6



(a) Periodogram

(b) autocorrelation

Fig. 3.6 Periodogram and autocorrelation to detect the period of user behaviour in reference region r_3, (a) periodogram of the pattern in r_3, (a) autocorrelation of the pattern in r_3

(b), the value in 24 hours is smaller than the value in 168 hours which is different from the graph inr_2. It shows that the strength of the user's 24-hour periodic behaviour is weaker than 168 hours which means that the user visits the region for fewer consecutive specific days per week. On the other hand, p_{23} is still larger than p_{13} because the number of samples for measuring 24 hours period is more than for 168 hours period.

In r_4, there is no valid period detected by ACF and DFT, however, in Fig.3.7 (a), we can find that nearly 95% of total timestamps are within the time from 17:00 to 19:00 every day.

Hence, if these timestamps are accumulated as a new three-hour interval and then use the combination of ACF and DFT with the period being 24 hours, it means that the user visits r_4 from 17:00 to 19:00 randomly every day (Fig.3.7 (b)).

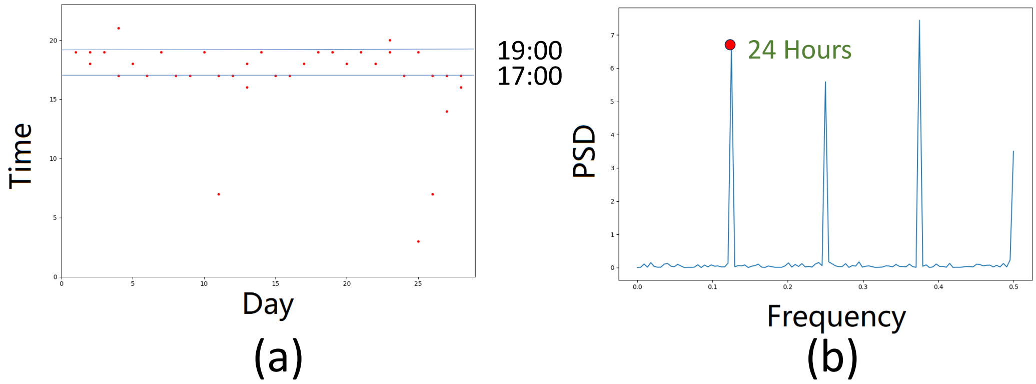


Fig. 3.7 (a)Timestamps in r_4 and (b)the periodogram of three-hour interval timestamps

Table 3.1 shows the period detected in four regions $r_1, r_2, r_3 and r_4$.

Table 3.1 The period in four reference regions

Reference region	r_1	r_2	r_3	r_4
Period	24H	24H,168H	24H, 168H	24H

3.3.4 POI preference establishment

In r_1, because the period is 24 hours, the system picks each interval from 0:00 to 23:00 in 28 days and uses Equation 10 to 12 to estimate the probability. And in r_4, the method is the same but the probability in the majority visit time 17:00, 18:00, and 19:00 is not high enough because a random visit in three hours for an hour disperses the probability. If a user spends two hours in a restaurant place from 17:00 to 19:00, the probability will be approximately over 50%which represents a two-hour duration visit. For the probability in r_2 andr_3, new probability $q^{'}$ needs to be considered due to two periodic behaviour. For the 24-hour cluster, we use 28-day data through the Bayesian algorithm and accumulate the resulting probability of 168 hours clusters. Because the occurrence day is mainly on weekday and weekend for r_2

and r_3 respectively, there are two summary probability distribution graphs for weekday and weekend which is Fig.3.8 and Fig.3.9.

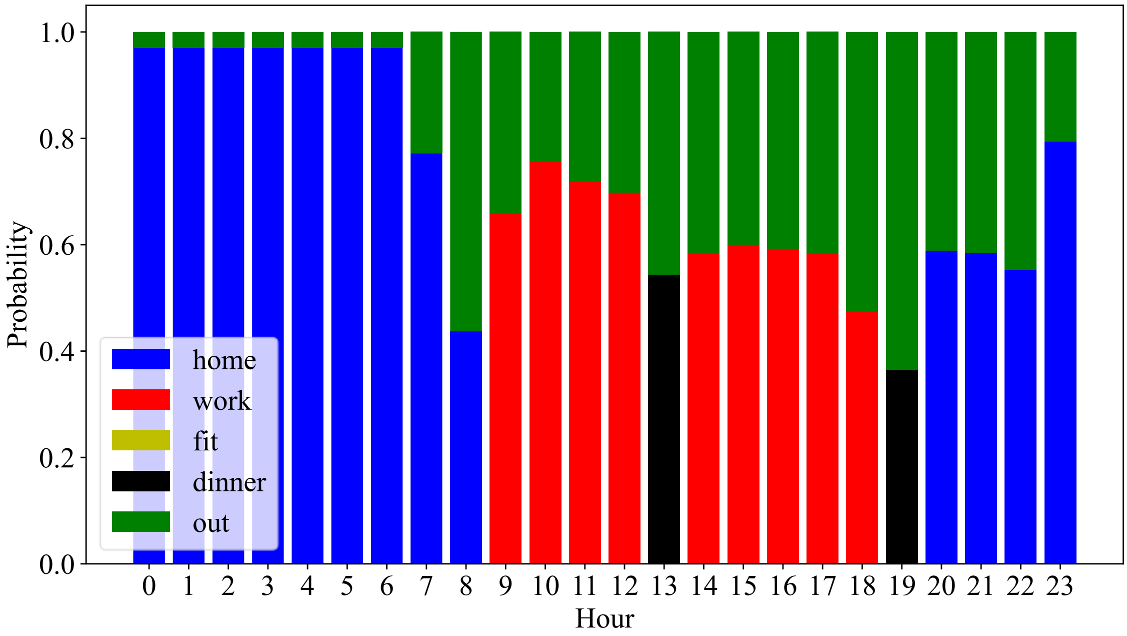


Fig. 3.8 Probability of movement on weekday

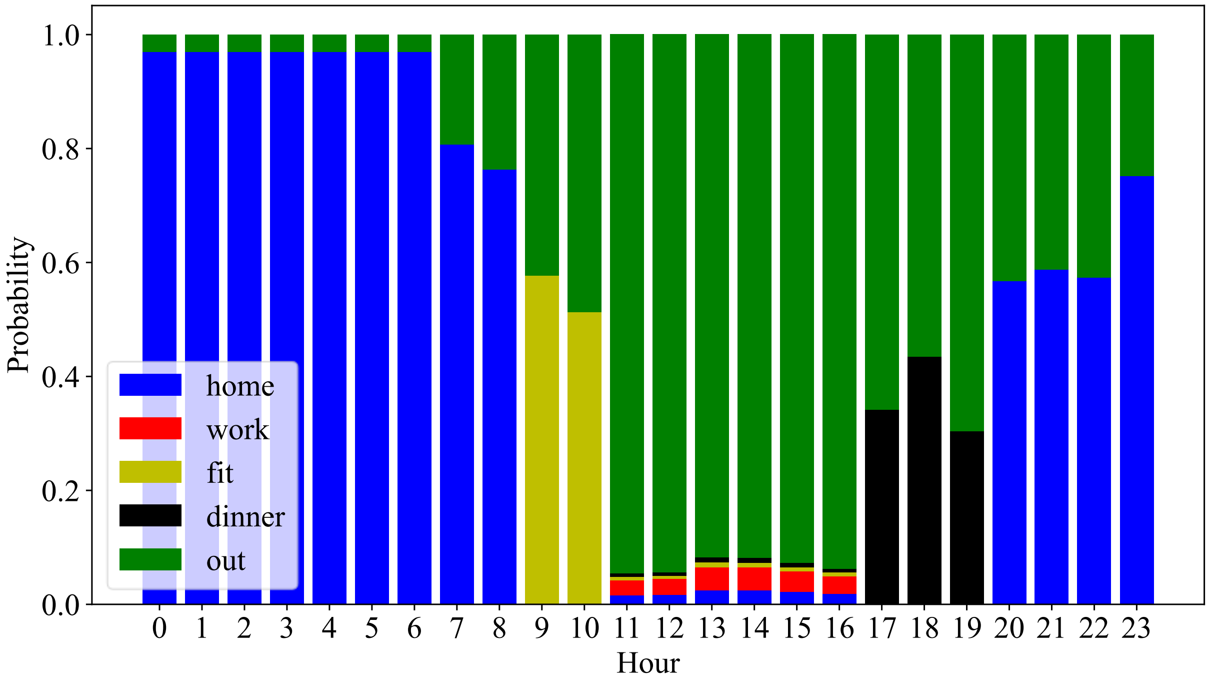


Fig. 3.9 Probability of movement on weekend

From Fig.3.8, it can be seen that the user leaves r_1 at around 9:00, return at about 20:00 and then has a high probability stay in this region until the next day at 9:00, which has a similar movement pattern with ‘home’ in the model. However, because of the influence of coming home later at weekends, the system reduces the probability of 20:00 to 22:00 on both weekdays and weekends. This is a problem for one periodic event which will be studied in future work. The user visits r mainly on weekdays from 9:00 to 16:00 which is the workplace in the model. r_3 is visited each Saturday and Sunday for two hours where is the gym. And r_4 appears from 17:00 to 19:00 every day with low probability where is the restaurant placed in the model. Therefore, the designed algorithm in this chapter can be used to detect the reference region, the period of visiting and the duration of stay in each period. In Fig.3.8 and Fig.**??**, the green bar represents the probability in the regions except the major region. For example, from 0:00 to 8:00, the major region is home (blue bar), and the sum of the probability of all of the other regions is the green bar because, in this chapter, the binomial distribution is proposed as likelihood to only estimate the probability of in or out the detected region. Thus, It cannot know the probability of visiting other regions from the perspective of the detected region. This problem can be solved by comparing the probability of all regions at the same time interval, then, choosing the region with periodic behaviour and the maximum value. Meanwhile, because of the probability density, even if all of the records are one (means in), the probability cannot be one hundred percent such as 0:00 to 6:00 for home.

Then, Personal POI is established. This list is divided into weekdays and weekends, respectively represented as $Q_{wd} = \left\{ q^{wd}_{j,1}, q^{wd}_{j,2}, q^{wd}_{j,3} ... q^{wd}_{j,i} | j = 1,2,...,5 \right\}$ and $Q_{wk} = \left\{ q^{wk}_{j,1}, q^{wk}_{j,2}, q^{wk}_{j,3} ... q^{wk}_{j,i} | j = 1,2,...,5 \right\}$, where $q^{wd}_{j,i}$ represents the possibility of j cluster at the ith hour on weekdays. Each list has a 5X24 dimension, that is, the possibility $q^{'}_{j}$ of each cluster location within 24-time intervals. As a result, the POI list of the past 28 days is established, and the corresponding regional clusters can be given for the subsequent locations, as well as the probability of each cluster in the corresponding time period.

3.4 Next destination Prediction Model based on POI Preference and LSTM

3.4.1 The Structure of the POI Preference Dataset

Although distinguishing between weekdays and weekends is clear for the analysis, there is a flaw in the prediction dataset preparation. The LSTM model performs sequence prediction, rather than input differentiation between weekdays and weekends. Therefore, to address this issue, we need to combine the probability lists for weekdays and weekends and introduce an additional dimension to indicate the weekday or weekend label. Base on that, we can merge both arrays into one cohesive dataset.

The final training dataset O is constructed as follows:

$$O = \{o_1, o_2, \dots, o_g\}$$
$$o_g = \{t_i, w_z, y_n, p_i^1, p_i^2, p_i^3, p_i^4, p_i^5\}$$

where t_i represents the time, including the hour, the day and the month. Since the time interval in this chapter is set at the hourly level, t_i will always be an exact hour. w_z refers to the day of the week, and z is a positive integer not greater than 7. y_n denotes the reference region information at time t_i, and n total clusters number, which is four. p_i^n represents the probability that the nth POI is visited by the user at time t_i. The values of p_i^n are aggregated from the $q_{j,i}^{wk}$ and $q_{j,i}^{wd}$ obtained in the previous section. Each o_g records the user's location, time, and probability of visiting each of the five regions at the current moment. Thus, we establish a probability matrix for each hour of the week, each reference region, indicating the likelihood of user visits, which will serve as the training set for the next destination prediction model.

The training and testing samples are split into 80% and 20%, which corresponds to 28 days and 7 days, respectively. Training dataset D is composed of segments, each consisting of l consecutive o_g elements. The historical window l refers to the number of time periods used as input to predict the next user visit location at time $g+1$. In this case, l is set to 12. Therefore, $D = \{(o_1, o_2, o_3, \dots, o_{12}), \dots, (o_{g-l}, o_{g-l+1}, o_{g-l+2}, \dots, o_g)\}$. The output data is the most likely POI $ylab_n$ for the next moment. It is important to note that p_i^n is actually calculated based on each time period within a week, so in practice, the historical data can be used initially, and then updated after a certain period of time to avoid real-time updating pressure on the system.

3.4.2 LSTM in Proposed POI Preference-based Destination Prediction

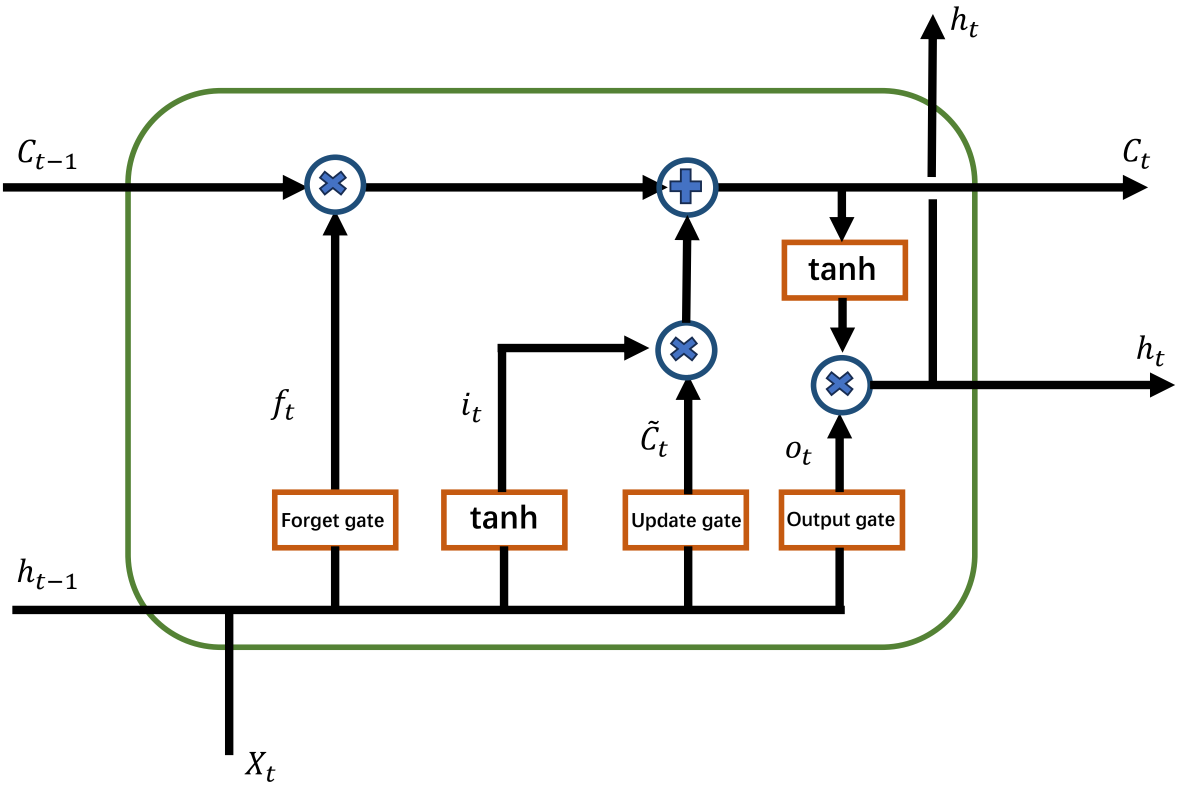


Fig. 3.10 The structure of an LSTM network

The main purpose of this chapter is to validate the improvement in the accuracy of predicting the next destination using the proposed POI preference, as well as to assess the predictive performance of the LSTM model on sequence data. The information included in the proposed POI preference dataset contains much information that goes beyond what traditional mathematical modelling or shallow machine learning algorithms can handle. For example, the MC transition matrix cannot fully describe the relationships between the possibilities of visiting different regions. It calls for a model that can extract features and validate the POI preference. LSTM, based on neural network models, can utilise cells to perform non-linear interactions with the inputs, allowing for the comprehensive exploration of features in complex elements. Additionally, its unique memory-forgetting mechanism can selectively retain information for a long time, which is crucial for sequence prediction tasks. Given these characteristics, this paper chooses LSTM as the model for predicting the next destination based on POI preference.

The functionality and characteristics of LSTM are worthy of research. Its basic architecture, RNN, achieves information retention by establishing connections between task-specific parameters in the hidden layers, which is a groundbreaking optimisation. This architecture has been widely used in natural language processing and sequence prediction tasks. These tasks emphasise that previous inputs can influence subsequent outputs, highlighting the importance of handling long-term dependencies. However, RNN suffers from the vanishing gradient problem [48]. From a mathematical perspective, this problem arises because the gradient updates exponentially decay to zero due to the increasing layers of backpropagation. From an information perspective, it occurs because the addition of historical features leads to feature dilution. LSTM, as an improved architecture of RNN, is designed to address these issues. The core purpose of LSTM is to replace the recurrent structure in RNN with gate structures called memory blocks [79]. These memory blocks control the efficiency of passing long-term information, effectively mitigating the problems of feature dilution and redundancy caused by the prolonged transmission of features.

Fig.3.10 shows the key components of LSTM, the memory blocks, which is a recurrently connected subnetwork comprising several functional modules called gates. During each iteration of the task, these memory modules perform non-linear computations on the features, allowing for the retention of important information and discarding less relevant information. This mechanism enables LSTM to selectively retain critical information for an extended period, thereby effectively addressing the vanishing gradient problem and handling long-term dependencies in sequence data.

The gates in LSTM are classified into three types based on their respective functions: the input gate i_t, the forget gate f_t, and the output gate o_t. Additionally, LSTM utilises a cell state C_t to store network state information from previous inputs. The functionalities of the gate structures can be summarised as follows:

- The forget gate, through its recurrent connection, controls how much information is retained in the current state of the memory block. The formula is as follows:

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \tag{3.13}$$

- The input gate controls the amount of new information flowing into the current state of the memory block. The formula is as follows:

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \tag{3.14}$$

- The representation of the cell state, controlled by both the forget gate and the input gate, is as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c) \tag{3.15}$$

- The output gate controls how much information is used to compute the output activation of the memory block, and its representation is as follows:

$$o_t = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \tag{3.16}$$

Where X_t is the input vector, h_t is the hidden state, W_{xf}, W_{hf}, W_{xi}, W_{hi}, W_{xc}, W_{hc}, W_{xo}, and W_{ho} are the weight matrices for each gate that need to be trained. b_f, b_i, b_c, and b_o are biases to better adapt the model. $\odot$ denotes the Hadamard product, $\oplus$ and $\otimes$ respectively represents element-wise addition and dot product of two vectors. σ and ϕ are activation functions, where σ is typically the sigmoid function and ϕ is typically the hyperbolic tangent (tanh) function. The choice of the sigmoid function is due to its output range from 0 to 1, which allows it to control the flow of information through gates and regulate the filtering of information. The tanh function is chosen to address the vanishing gradient problem, as it can sustain long-term second derivative functions before approaching zero. The sigmoid and tanh functions are defined as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.17}$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{3.18}$$

The execution process of LSTM involves the previous hidden state h_{t-1} and the input value X_t at the current time step. These two parameters are first passed through the forget gate to determine the information to be discarded, resulting in f_t. Next, they are passed through the input gate to determine the information to be updated, resulting in i_t, as well as the current cell state $\widetilde{C_t}$. Finally, the outputs from the forget gate and input gate are combined to obtain the long-term memory C_t and short-term memory h_t, respectively. These values are then stored and used as input for the next neuron.

3.4.3 Baseline algorithms

The innovation of this chapter lies in the exploration of user POI preference by leveraging their historical location and time information. By combining spatial and temporal aspects, we establish a framework that considers each POI's spatial characteristics as a reference point, thereby independently investigating the temporal dependencies of different POIs. Moreover, we construct a training dataset for POI preference, which includes user visit probabilities, to better guide the sequence neural network for the next destination prediction. Therefore, there are three aspects to be addressed to demonstrate the superiority and interpretability of the proposed method: (1) comparison of data POI extraction methods, (2) comparison of training data structure, and (3) comparison of different model parameters.

For the comparison of data POI extraction methods, this chapter refers to state-of-the-art techniques to validate the reliability of establishing preferences. The Periodica [162] and

DBSCAN models [54]are introduced. The former method involves constructing a bivariate normal density map to select POIs and then using maximum likelihood estimation to model the visit probabilities of these locations, which is similar to the type of training data this chapter is constructing. The latter method relies on taxi driver patterns to extract POIs, further demonstrating the strong performance of DBSCAN in the context of POIs. Therefore, for data POI extraction, both of these methods will be used. The former method can be applied to the training model, while the latter can be used to perform reference region extraction by masking the KDE proposed in our model.

For the comparison of data types, we will compare the accuracy of predicting the next destination by establishing training datasets with different elements. Currently, most research focuses on processing POI geographical locations and sequences, which includes only geographical location information and time information, and focuses on the length of historical movement trajectories. Studies based on MC [55] and advanced models like HST-LSTM [49]have used these types of data. There are also studies that have added time and distance dimensions [56], calculating the time and distance between two points of visit and incorporating them into the training data construction.

Finally, it will investigate the impact of self-parameters in this chapter, focusing on the top-$p\%$ in KDE on the prediction results. We will choose different values of p, such as 20, 40, 60, 80, and 100, for analysis. When p is set to 100, it is equivalent to using DBSCAN alone for reference region localisation.

Therefore, this chapter will use the following models and training data types as baselines to demonstrate the advantages of the proposed framework in prediction:

- Periodica-LSTM [162]. It is a hybrid approach that combines the advanced Periodica algorithm, which calculates visit probabilities, with LSTM for prediction. While Periodica was initially designed for detecting periodicity in bird movement data, it lacks a predictive mechanism for users. Therefore, in this chapter, as a baseline method, LSTM is integrated into the framework to enable destination prediction based on the calculated visit probabilities.

- DBSCAN-only model [54]. As DBSCAN is still considered a crucial technique for POI extraction, this study verifies the significance of fine-grained POI localisation in the proposed method by excluding the KDE component. In the DBSCAN-only model, the DBSCAN algorithm is solely used for extracting reference regions, without considering the KDE probabilities.

- FPMC [55]. FPMC is the state-of-the-art personalised Markov Chain algorithm for recommendation or prediction tasks.

- HST-LSTM [49]. This model is an improved version of LSTM, which incorporates historical data through an encoder-decoder architecture to enhance prediction performance.

- LSTM. To validate the impact of incorporating probabilistic data, this method directly uses LSTM to predict the next destination based on historical POI and time data.

- STGCN [56]. This model enhances the prediction performance by incorporating contextual information, such as the distance between POIs and time, into the data. The predictor used in this model is LSTM.

- p=20,40,60,80,100 refer to the proposed model in this paper, which select the region covered by the top 20%, 40%, 60%, 80%, 100% probability distribution in KDE (Kernel Density Estimation), respectively.

By comparing the results, the contribution degree of each POI model method to the prediction algorithm can be evaluated. It has been demonstrated in the literature review that such strongly correlated auxiliary information can be used to enhance system performance.

3.4.4 Model Configuration

In the LSTM-based model, the length of historical data is set as 12. LSTM has two layers, and the number of hidden units is set as 16, 32. All models take sequential historical data as input and output the POI label for the next time step.

By separately training the data of the first 28 days of the LSTM, It can then predict the testing data using the next moment prediction, and the accuracy rate is used to evaluate the performance. In addition, since the data in the early morning (00:00-06:00) is mostly fixed at home, although it is brought into training and prediction for the dataset completeness, the result is not included in the accuracy calculation.

To assess the performance of the models, predictive accuracy is applied, which is the proportion of correct predictions to total predictions made. The accuracy can be defined as follows:

$$Accuracy = \frac{1}{I}\sum_{i=1}^{I} L_{\hat{ylab}_i=ylab_i} \times 100\% \tag{3.19}$$

, where I is the total number of testing data. $L_{\hat{ylab}_{i_i}=ylab_i}$ represents the number when ith timestamp, the POI prediction $\hat{ylab}_i$ and real data $ylab_i$ are the same, which is 1.

3.4.5 Prediction Performance

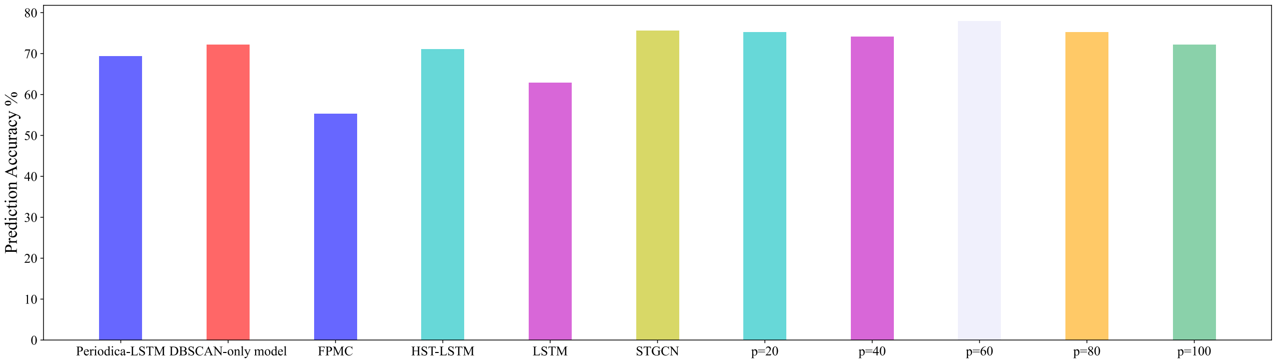


Fig. 3.11 The performance of the proposed model and baseline models

Fig.3.11 illustrates the prediction performance of the proposed model and the baseline model.

In order to facilitate comparison with other models, the most suitable value of p is selected as $p = 60$. It can be observed that some points clustered as POIs by DBSCAN may have a negative impact on predictions. Additionally, as p values approach the minimum and maximum, the model's performance tends to decline. This is in line with common sense, as too small a range may disrupt the continuity of visitation records, while too large a range may be affected by noise points. Another interesting phenomenon is that the model's performance at $p = 40$ is worse than at $p = 20$. This could be due to 40% of the region being in an intermediate zone with partial periodic points. Whether below 40% or above it, the periodicity becomes clearer. This suggests that it may be related to the user's periodic behavioural patterns. Overall, the system performs best when $p = 60$, with a performance improvement of 7% compared to the worst case of $p = 100$. This indicates that KDE effectively discards the edge points of DBSCAN, thereby improving overall performance. This validates the necessity of researching from the perspective of spatiotemporal dependence.

Next is the study of the impact of POI preference extraction quality on predictions. Firstly, there is a clear difference between Periodica-LSTM and the proposed model, which demonstrates that our proposed model has a better ability in extracting mobility patterns. This is evident when comparing with the DBSCAN-only model as well. Regarding the comparison of training data assistance, it can be observed that studying historical visitation sequences alone may have reached a bottleneck. For instance, in the comparison between LSTM and the proposed model, there is nearly a 20% difference due to the variation in training data dimensions. The same situation is observed in the comparison between Periodica-LSTM and the LSTM model. Although Periodica-LSTM's information feature extraction efficiency is lower than ours, it still outperforms LSTM in the models that only focus on historical visitation sequences. Furthermore, even with algorithm optimisation, it is difficult to further extract features at a deeper level. The most obvious example is HST-LSTM, which, despite using a deeper neural network architecture, does not yield significant gains in exploring historical visitation data compared to the other two models that involve auxiliary information, STGCN and the proposed model. After comparing the two models that incorporate auxiliary information, it is evident that the information regarding user visitation probabilities, proposed in this study, leads to a more significant improvement in prediction performance, with an increase in accuracy of approximately 4%.

Another essential point to note is the significant difference between the MC model and neural network-based models. This can be attributed to the advantages of LSTM/RNN in extracting sequence features. As seen in the comparison between LSTM and HST-LSTM, model optimisation can indeed lead to performance improvements. Furthermore, with the assistance of auxiliary information, data can be more effectively utilized to enhance the performance of the system model. Therefore, performance optimisation and intrinsic feature extraction from data are the two key factors in improving model performance.

In summary, auxiliary information can bring surprising improvements to neural network-based models in sequence prediction tasks. The framework designed in this chapter, which utilises LSTM for next destination prediction through POI preference, achieves a prediction accuracy that is 4% higher than the state-of-the-art algorithms and approximately 30% higher than traditional methods.

3.5 Conclusion

This chapter presents a framework that leverages spatiotemporal dependencies to explore users' POI preferences, resulting in auxiliary information for constructing the training data. By applying this training data, our proposed model exhibits enhanced efficiency in feature extraction compared to traditional historical geographical coordinate data. Through validation on real-world datasets, the developed system achieves a 4% improvement in prediction accuracy over state-of-the-art algorithms and outperforms systems relying solely on traditional historical geographical coordinate data by 20%. This is a satisfying outcome, indicating the feasibility of utilizing user-specific POI data to enhance the detection of periodic behaviours.

Furthermore, individual POI preferences can also serve as location information. For instance, when the highest probability region remains consistent over a period of time, the user is likely to stay at the same place. By exploring the time of the next highest probability region, the duration of their stay can be determined. This can be beneficial for other network services, such as network prediction and device-to-device (D2D) communication, which require devices to maintain relatively stable states.

Furthermore, this study also verified that the addition of auxiliary information can greatly assist in feature extraction for neural network-based sequence prediction models, thereby enhancing prediction performance.

The proposed model has two limitations. Firstly, it lacks a mechanism for parameter adjustment. For instance, as the data volume increases, selecting appropriate KDE parameters becomes a challenging task. Employing exhaustive search algorithms to find the optimal value of top-p becomes impractical when the number of POIs grows substantially. To address this issue, alternative approaches such as Bayesian parameter tuning or using neural network frameworks for parameter selection could be introduced. Secondly, the model overly relies on the quantity of historical data for periodicity detection, which is a common challenge in statistical algorithms. Typically, resolving such problems involves introducing neural networks for nonlinear modelling, but this could potentially increase complexity.

In conclusion, the proposed algorithm effectively establishes personalised POI preferences and generates auxiliary information to facilitate the construction of state-of-the-art sequence prediction models. The algorithm demonstrates strong predictive capabilities, showing significant improvements over traditional approaches and current advanced methods. Overall, it successfully enables the establishment of personalised POI prediction and offers valuable insights for the development of sequence prediction models.

Chapter 4

MTL-STPN: a Multi-Task Learning-based Spatial-Temporal Parallel deep learning Network

In the previous chapters, the potential of deep learning models in time series prediction tasks for mobile networks has been discussed, as well as the importance of feature exploration based on auxiliary information to optimise the performance of prediction models. According to these achievements, this chapter proposes a novel framework for mobile network traffic prediction that combines deep learning techniques with the spatiotemporal correlation of mobile network traffic. This chapter introduces a Multi-Task Learning-based Spatial-Temporal Parallel deep learning Network (MTL-STPN) that is able to learn nearby highly correlated mobile traffic by sharing parameters. Complemented by a traffic pattern clustering method, Spatial-Temporal Pearson traffic Pattern Clustering (STPPC) algorithm, the framework can autonomously select highly-contributing base station traffic to maintain the model's robustness. Numerical results demonstrate that MTL-STPN outperforms existing prediction algorithms in cell-level mobile traffic prediction tasks. Furthermore, the multi-task prediction in MTL-STPN enhances the learning efficiency of base learners, reducing the required training period for parameter training by approximately 60%, while maintaining similar or better prediction performance.

4.1 Introduction

Over the last decade, the dramatic development of cellular networks has led to the explosive demands of mobile traffic. In order to reduce OPEX while guaranteeing the QoS, operators tend to design a more intelligent and self-organising system, called the SON. It introduces demand-aware allocation mechanisms of network resources to cope with the fluctuations in data usage and take preemptive actions, such as Radio Resource Management (RRM), BS sleep-mode strategy, etc. The operating of these strategies is based on the analysis of real-time traffic and accurate prediction capabilities [174], which are the major challenge to implement. Therefore, understanding and forecasting the mobile network traffic patterns are the basic information in the self-organising system to alleviate the burdens on the network operator's infrastructure caused by the massive data transmission, which is valuable from the perspective of business development as well as user experience.

The short-term prediction of local aggregate mobile internet traffic is an essential subject in the research. Since such data has strong diurnal patterns and periodicity, time series analysis and machine learning algorithms have been widely used to model and predict mobile network traffic. Moreover, to further improve the accuracy of short-term data traffic prediction, the features of spatiotemporal correlation are extracted inherently. The mobile traffic in a target cell for the next timestamp forecasting has partially correlated with the historical traffic and current network traffic both in the target cell and adjacent cells. In the early research, [175] utilized Conditional Entropy (CE) theory to quantify the predictability of the traffic under three types of services and concluded the positive correlation between the knowledge of adjacent cell traffic and the target cell's predictability. [176] verifies this conclusion and further shows that similar attribution among different regions, like clustering by POI, has a positive influence on the result of cellular network traffic prediction in its framework.

Due to the complexity of spatiotemporal feature modelling, deep learning-based approaches have been studied recently including customising the deeper models to fit various traffic patterns or increasing the receptive field to extract more features in historical data. For the former method, for example, [177] employed a hybrid deep learning model, including the global and local autoencoder-based model and LSTM units, to achieve the purpose of spatiotemporal features modelling. Moreover, the densely connected CNN has also been applied to extract the spatiotemporal dependencies by treating data traffic snapshots as images[178]. Furthermore, ConvLSTM is a recently popular deep learning-based method while considering both characteristics of temporal sequences and spatial correlations[179]. Compared with customising models, increasing the receptive field, or applying other related types of data, can also get improved results without increasing the complexity of the system. In [180], by using a stacked structure named Spatial–Temporal Cross-domain neural Network (STCNet), different types of cellular traffic can be investigated simultaneously to improve the accuracy of the results. A multi-input framework called Deep Traffic Predictor (DeepTP) is also proposed in [176], which aimed to discover the unique property of the traffic pattern in the similar POI and time effect.

However, as the application scenarios become more complex and the observation granularity increases, an increasing number of composite pattern traffics are emerging. Although their traffic still exhibits relative periodicity, the traffic patterns in different regions are showing significant variations, which has already garnered some attention [57]. Overall, prediction models need to be robust in predicting different traffic patterns while efficiently identifying highly correlated traffic to enhance prediction accuracy. Therefore, multi-task learning, as a promising approach that enables joint learning of different tasks through parameter sharing, attracts some attention to improve performance. Through jointly introducing the related tasks, different features are extracted to obtain more effective information for respective predictions. [181] discusses the impact of different multi-task structures on the RNN-based model and concluded that the multi-task learning model can indeed improve performance. Meanwhile, a novel MTL architecture is proposed which has the shared and dedicated learning layers separately. It is beneficial to generate and exploit common information while maintaining the effect of specific task features.

At present, there is a lack of research on the impact of the correlation between tasks on the results of MTL prediction in the mobile traffic forecasting field. Most of the work directly apply the nearest N areas as N tasks [181][182]or treats the distance of the target cells as an essential measurement factor[183]. It limits the scope of task selection. Meanwhile, due to the different attributes of urban areas, it is possible that nearby low-correlation areas are included, which will degrade performance. Besides, although the single-task learning model can be multi-tasked to achieve the purpose of spatiotemporal dependencies extraction, it cannot avoid that the convolutional-based STL model takes its advantage on spatiotemporal correlation modelling in traffic prediction.

To fill the aforementioned gaps and address the new challenges in network traffic prediction, a novel deep learning structure named Multi-Task Learning Spatial-Temporal Parallel deep learning Network (MTL-STPN) is proposed. By deploying proposed framework in the cloud, it enables real-time monitoring of city traffic and enhances short-term aggregated mobile traffic prediction performance through the extraction of spatial and temporal features from multiple regions.

LSTM and 3D convolutional neural networks are connected in parallel to model the temporal characteristics of the target cell and the spatiotemporal characteristics of neighbor cells, respectively, and then to predict the traffic of the target cell at the next timestamp through a fully connected network. In the MTL framework, each related task shares the LSTM layer to improve the effectiveness of temporal features capturing. Moreover, according to clustering traffic patterns in the target region, which is composed of target cells and adjacent cells, a specific region correlation detective method based on temporal and spatial dependencies is designed to label the relevant regions as tasks for MTL-STPN. By filtering out low correlation regions, the efficiency of feature modelling is raised while reducing the complexity of the model.

There are four main contributions of this study:

- A novel LSTM-3D CNN model is proposed as STL-STPN. The STL-STPN model combines two algorithms in parallel to capture distinct features: the CNN is employed to explore spatial features of neighboring traffic, while the LSTM model is utilised to model the time series traffic, effectively capturing both spatial and temporal contextual information. The two feature representations are then fused using element-wise addition to achieve feature integration. Finally, a multilayer perceptron (MLP) is applied to model the fused features, thereby uncovering the spatiotemporal correlations. Experimental results demonstrate that this model outperforms the state-of-the-art CNN-LSTM model, leading to improved performance measured by root mean squared error.

- A novel MTL framework named MTL-STPN is proposed, which is combined by multiple STL-STPNs, to further improve the accuracy of traffic prediction. By sharing the parameters of the temporal feature extraction layer of the MTL model, and then combining them with the spatiotemporal correlation features extracted by the respective task dedicated learning layer for prediction, the accuracy can be raised while reducing the model convergence time.

- A task correlation measurement algorithm named Spatial-Temporal Pearson traffic Pattern Clustering (STPPC) algorithm is designed to fit the MTL-STPN model. Based on the classification of different traffic patterns in the target multi-cell and the central cell traffic similarity measured by the Pearson correlation coefficient, the potential MTL regions can be sorted.

- The performance of MTL-STPN utilising real-world cellular network traffic traces collected in Milan is evaluated. Numerical results will validate the advantage of the MTL-STPN compared with other baseline methods in terms of prediction accuracy and time consumption.

The remainder of this chapter is organized as follows. Section 4.2 describes the open dataset from Telecom Italia applied in this chapter and presents the statistical characteristics of the dataset. Section 4.3 elaborates the structure of MTL-STPN. Section 4.4 introduces the detail of the STPPC algorithm. Section 4.5 shows the conducted results of MTL-STPN and other baseline methods. Section 4.6 gives the conclusion of this chapter.

4.2 Data and challenges

4.2.1 Description of the mobile network traffic trace

In order to validate the performance of the proposed model, the integrity of the dataset is strictly required. In this experiment, the dataset was collected from Italian telecom in 2015, which has been widely used in big data analysis projects. It consists of call detail records (CDR), internet, weather, and social networks in Milan and the Province of Trentino during November and December 2013. The geographic location of the data matches with standard WGS84 (EPSG:4326). Internet data in Milan is applied to evaluate the performance of the proposed model in this chapter.

The data is collected by meshing Milan into 100×100 grids of the same size, and separately recording the aggregated internet traffic in each cell with a temporal interval of 10 minutes. The size of each cell is 235×235 meters. In this article, I set the coordinate axis whose origin is at the upper left corner of the whole map to calibrate ID. The data includes ID, timestamp, and the aggregated traffic in each cell. The time interval is adjusted to hour by averaging the original data in the one hour period. The spatiotemporal sequence of the data matrix is $D = (Dt|t = 1,2,...,T)$, where T is the number of timestamps. D_t represents the traffic snapshot matrix at timestamp t and the grid overlay L×W (100×100) cells, which can be written as:

$$D_t = \begin{bmatrix} d_t^{(1,1)} & d_t^{(1,2)} & \cdots & d_t^{(1,W)} \\ d_t^{(2,1)} & d_t^{(2,2)} & \cdots & d_t^{(2,W)} \\ \vdots & \vdots & \ddots & \vdots \\ d_t^{(L,1)} & d_t^{(L,1)} & \cdots & d_t^{(L,W)} \end{bmatrix}$$

Where $d_t^{(l,w)}$ is the traffic volume in the cell (l,w). The dataset can be denoted as a spatiotemporal sequence with the tensor $D \in \mathbb{R}^{\mathbb{T}\times\mathbb{L}\times\mathbb{W}}$. Besides, the traffic flow at cell (l,w) can be represented as $f_{tc}^{(l,w)} = \{d_t^{(l,w)}|t = 1,2\ldots T\}$.

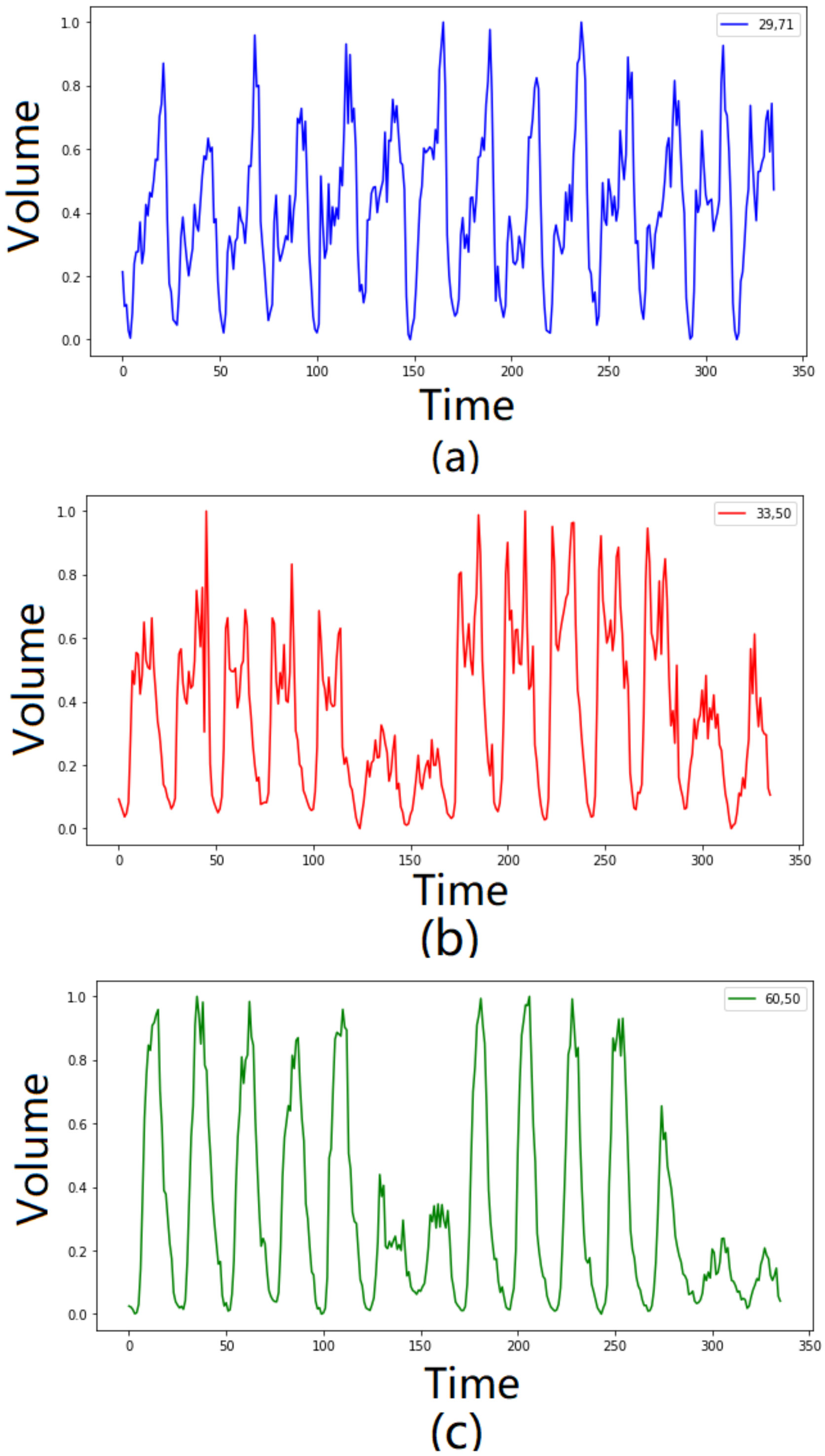


Fig. 4.1 Three typical traffic patterns in Milan dataset

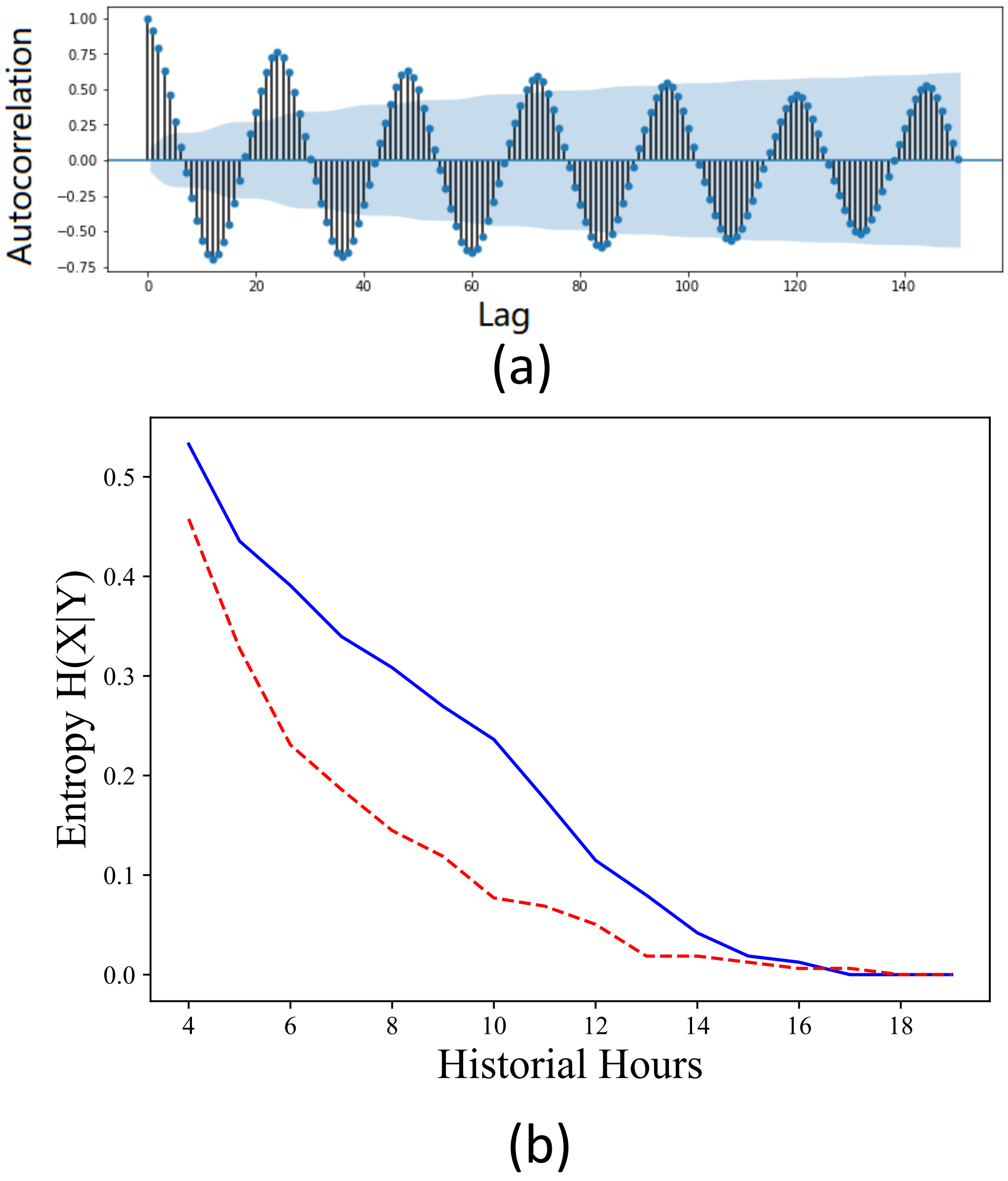


Fig. 4.2 (a)ACF of a typical cell's traffic (b)Spatial CE in traffic between adjacent cells and high-correlation

In this subsection, Milan mobile traffic dataset is analysed in traffic patterns perspectives and spatiotemporal dependencies, respectively. It depicts that traffic pattern has significant difference depends on its location. However, from the temporal domain, most aggregated traffic remain periodic patterns. Moreover, the traffic pattern relation between short distance

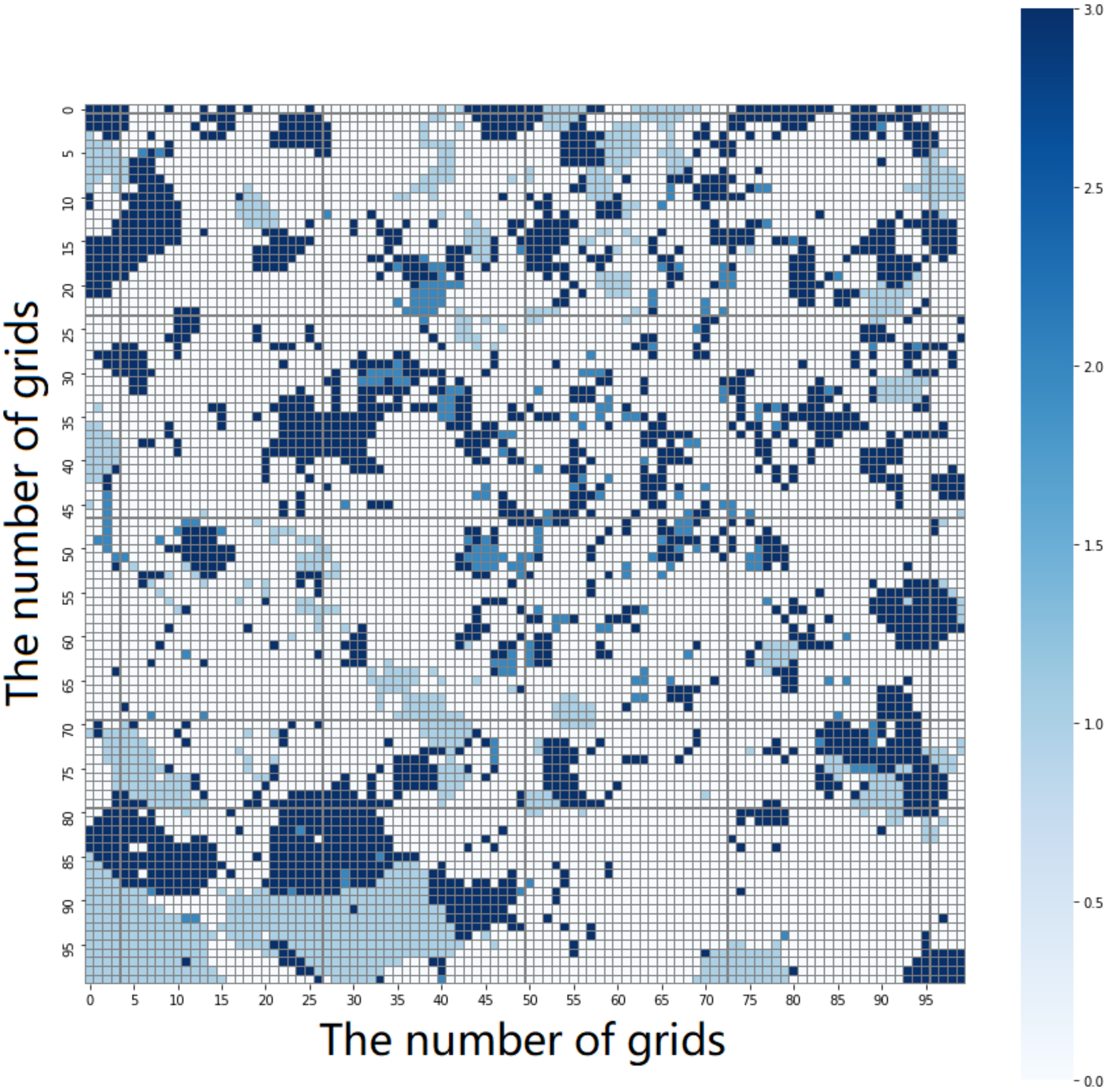


Fig. 4.3 Pearson correlation value between a traffic in a target grid and other traffics in all the grids

cells and long distance cells are demonstrated by spatial CE and Pearson coefficient. From Fig.4.1, Fig.4.2 and Fig.4.3, there are some observations obtained.

Observation 1: Fig.4.1 demonstrates the two-week-long traffic patterns of three typical cells, respectively. All these traffic illustrate periodical daily patterns. Besides, it is obvious that the traffic of some cells changes according to weekdays and weekends, but others remain the same. Auto-correlation coefficients of the traffic are shown in Fig.4.2 that they all have non-zero auto-correlations in the temporal domain which means that the traffic is predictable through historical data. However, the pattern categories of different cells are different, especially between weekdays and weekends, which may affect the forecasting performance.

Observation 2: Fig.4.3 shows the correlation between a traffic of specific cell and the traffic of other cells in terms of Pearson correlation coefficient ρ, which is a widely used tool in measuring spatial correlations [31]. It indicates the positive and negative correlation by the sign of the number, and the magnitude indicates the similarity between the two traffics. As can be seen from the figure, the correlation of temporal traffic does not only depend on the distance between two cells. Even if far away from the target cell, there are quite a number of cells that have similar traffic to the target cell has.

Observation 3: Fig.4.2(b) compares the spatial CE value calculated by the traffic of a target cell and its high correlation cell and its adjacent cell, respectively. The data is divided into ten levels and then calculate the CE which is the same as in chapter . It can be found that the CE value of the similar pattern set is lower than that of the adjacent cell set, which means that there is not a complete negative correlation between distance and predictability. In other words, the distance between the cells cannot be a reliable standard in the prediction. Simply enlarge the spatial range of the observation region may bring a negative influence on results because of such low-correlation patterns. Spatial and temporal correlations should be considered simultaneously to enhance the performance of the model.

Based on the above analysis of the spatiotemporal correlation between the cells, it can be concluded that spatiotemporal features should be involved to help the improvement of prediction accuracy while the similarity in the traffic pattern should be prioritised instead of the distance.

4.3 Methodology

This section first describes the structure of a spatial-temporal parallel deep learning model and its components, and then builds an MTL-STPN model with shared-dedicated structure for mobile network flow prediction. Finally, the STPPC algorithm is proposed to select the strong correlation features between various prediction tasks.

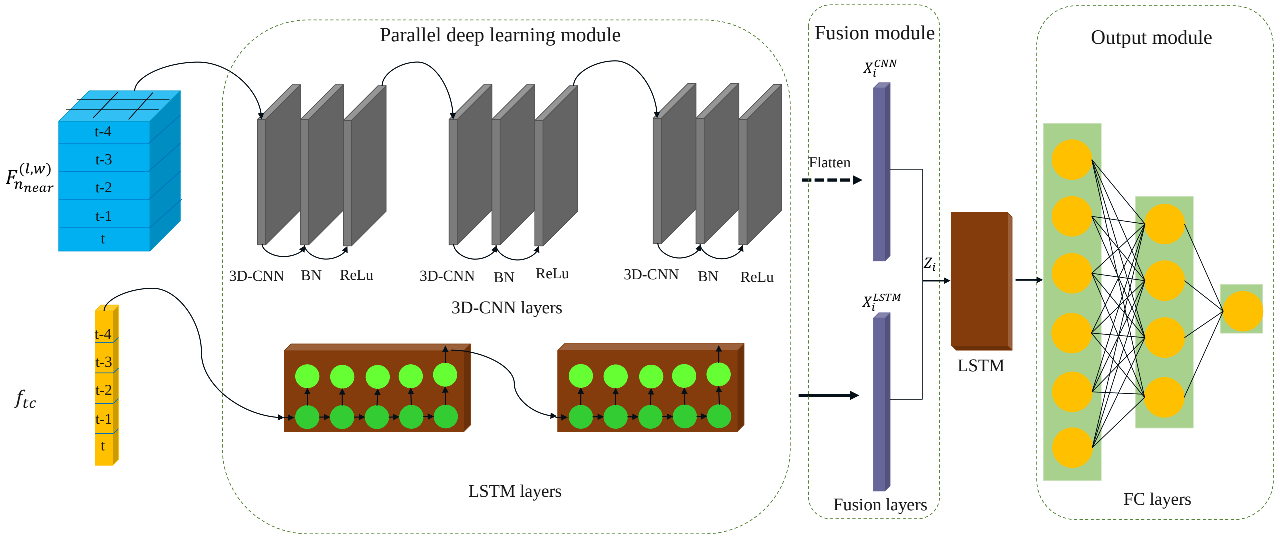


Fig. 4.4 The structure of STL-STPN

4.3.1 Spatial-Temporal Parallel deep learning Network

A deep neural network architecture, called Spatial-Temporal Parallel deep learning Network (STPN), is proposed to apply in aggregate mobile traffic forecasting. The proposed STPN contains a parallel structure of LSTM and 3D-CNN model, as shown in Fig.4.4.

The goal of this chapter is to predict the specific cell traffic. The traffic in the target cell (l,w) represents as $f_{tc}^{(l,w)}$, which is a traffic volume sequence with T elements. Target region locates the target cell as the centre grid, and the region composed of $n_{near} \times n_{near}$ adjacent cells. Therefore, its traffic is a three-dimensional matrix $F_{n_{near}}^{(l,w)}$ with the size of $T \times n_{near} \times n_{near}$, represented as

$$F_{n_{near}}^{(l,w)} = \begin{bmatrix} f_{tc}^{(l-\frac{(n_{near}-1)}{2},w-\frac{(n_{near}-1)}{2})} & \cdots & f_{tc}^{(l-\frac{(n_{near}-1)}{2},w+\frac{(n_{near}-1)}{2})} \\ \vdots & \ddots & \vdots \\ f_{tc}^{(l+\frac{(n_{near}-1)}{2},w-\frac{(n_{near}-1)}{2})} & \cdots & f_{tc}^{(l+\frac{(n_{near}-1)}{2},w+\frac{(n_{near}-1)}{2})} \end{bmatrix}$$

Although it has been proved that distance and traffic correlation have no significant relation, the spatiotemporal dependencies still need to be considered because the traffic in the adjacent cells indeed usually exhibits a strong correlation. Therefore, in this article, the value of n_{near} is set to be small, like 3 or 5, which means 3 × 3 grids or 5 × 5grids. n_{near} must be an odd number and bigger than 1, which is symmetrical. On the other hand, small scope of the observation area (input data) can help to reduce the computation of the model.

By cooperating with the characteristic of two algorithms in temporal and spatiotemporal feature extraction, STPN can be exploited for the further improvement of mobile traffic forecasting compared to both algorithms. In this case, LSTM and 3D-CNN are fed with target cell traffic $f_{tc}^{(l,w)}$ and target region traffic $D_{n_{n}ear}^{(l,w)}$, respectively. Then reconstruct the data structure to fuse and return these features. The output is fed into a new stacked LSTM network to further capture the temporal dependencies. In the end, the MLP generates the prediction value through fully connected layers. The following subsections describe the operation of each component.

LSTM: The recurrent neural network has been widely used in handling time sequence forecasting problems. With the observation window recursive slide, the output from the previous state is fed into the next state with the information captured from the internal correlations over time. When the recurrent finished, the output is considered to be a summary through all past observations, which is the prediction result generated. LSTM is a special kind of RNN, that has been explicitly designed to avoid the long-term dependency issue, which is the cause of the vanishing gradient problem in normal RNNs [184]. The capability of learning long-term dependencies is from the delicate control gates as the basic units in LSTM. A standard LSTM consists of a memory cell c_t, input gate i_t, forget gate f_t, and output gate o_t. The memory cell will be updated when the forget gate and the input gate decide which information should be abandon or save. Then the output is generated by pointwise operating the updated memory cell value and the output gate value as the new hidden state and the result. With the help of gates operating, LSTM can extract the long-term temporal dependencies of mobile traffic.

In STL-STPN, multiple layers LSTM units are stacked as an LSTM network, that aims to precisely capture the temporal features of the mobile traffic. It can be pre-set the number of features in each layer passed to the next layer by controlling the number of units in LSTM. The depth of the whole network (e.g., the number of layers) will influence the performance and the efficiency of the mode. For the next-timestamp prediction, the model observes the historical traffic with a fixed size TS, which is hyperparameter, then predicts the traffic at the next timestamp T + 1. Therefore, the training inputs can be set as a TS + 1 array, which means TS observation timestamps and a label. The data has been normalised to optimise the convergence process.

3D-CNN: Differ from LSTM in the perspective of temporal modelling that can retain long-term information, 3D-CNN does not enable to have back-propagation through time (BPTT), which reduces the representatives in single time series feature extraction. However, the characteristic of the convolutional layer can enable model multi-temporal traffic to capture the cross-temporal correlations by sharing the weights across different locations in the input. It can enhance the detecting capability of minor fluctuations in temporal through spatial dependencies. Besides, batch normalisation (BN) is usually added before the activation function to normalise the input value of the activation function, to solve the impact of offset and improve the generalisation ability of the network.

The inputs can be set as a $TS \times n_{near} \times n_{near} \times 1$ four- dimensional matrix, which means TS observation timestamps, $n_{near} \times n_{near}$ cells and 1 type of data (channel), which the traffic volume is normalised.

The convoluted feature maps $M_j^c = \text{LeackyReLu}\left(\sum_{n=1}^{N} M_i * W_{ji} + b_j\right)$ with a sequence i of spatiotemporal inputs $M_i | i = 1, 2, 3...$ * is the convolutional operator and in this research, Leacky ReLu is applied as activation function.

Fusion layer: To enhance the capability of spatiotemporal features extraction and prediction performance, a fusion layer is added to blend the features captured by both models. It enables to establish a kind of ensemble system with two dedicated deep learning-based algorithms to take advantage of both sides, which has been proven in the improvement of prediction accuracy [177][174]. In this case, both target cell traffic temporal features (by LSTM) and its adjacent cells traffic spatiotemporal features (by 3D-CNN) are jointly exploited to improve the accuracy of the forecasting, which helps to exceed their individually employing performance, respectively. A problem worth noting is that two types of spatial information will reduce the effectiveness of feature capturing and weight influence, especially when the prediction target is a single traffic. ConvLSTM is replaced by the LSTM algorithm to reduce the complexity of the whole model while maintaining the temporal information concern. Meanwhile, 3D-CNN is retained to continuously monitor mobile traffic fluctuations. Moreover, a stack LSTM network at the end of the fusion layer helps to leverage the temporal features. The inner feature matrix captured from 3D-CNN and LSTM will both be reshaped the construction to $TS \times n_{near}^{2}$ for element alignment. In the fusion layer, element-wise addition M^{f} is applied as follows:

$$M^{f} = \text{LeackyReLu}\left(\text{add}(O^{LSTM}, \text{reshape}(O^{3DCNN}))\right)$$

where O^{LSTM} and O^{3DCNN} are the features extracted from LSTM and 3D-CNN.

4.3.2 Multi-Task Learning STPN

In [183], it has been proved that the prediction performance enables to be enhanced by jointly training multiple relevant tasks. In this section, an MTL model based on STPN is proposed as MTL-STPN, which aims to improve the performance of the prediction task.

According to the meaning of target cell and target region, reference cell is the location that its traffic flow could be applied in the MTL-STPN as one of the tasks. The traffic in the reference cell (l', w') represents as $f_{rc}^{(l',w')} = (f_{rc,1}^{(l',w')}, f_{rc,2}^{(l',w')}, \cdots, f_{rc,T}^{(l',w')})$, which is a traffic volume sequence with T elements. The spatiotemporal traffic matrix in reference region constructed by reference cell (l', w') and its adjacent cells, represents as $F_{n_{near}}^{(l',w')}$.

Since STL-STPN is a parallel structure, the layer on one side can be allocated as the shared layer, as shown in Fig.4.5. In this study, the LSTM layer is used as the shared layer. Due to the complicated spatial distribution of traffic, it requires multi-region features are extracted efficiently. Sharing the temporal feature representations generated by other

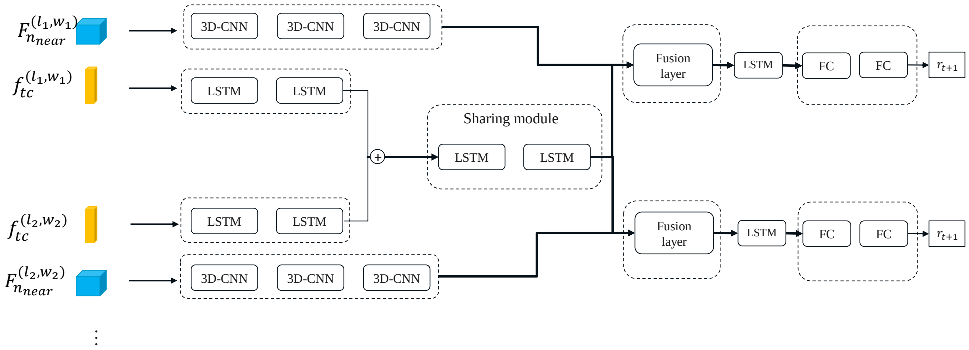


Fig. 4.5 The structure of MTL-STPN

tasks' LSTM module can help in the development of spatiotemporal feature, which has introduced in chapter . Compared to single task ConvLSTM algorithm, the efficiency of the model is increased by extracting the spatial features from the LSTM part, and the amount of calculation is not increased significantly. For 3D-CNN as a dedicated layer, the shared layer can not only provide new feature maps of spatial distribution, but also make up for its weak feature of mode ling long sequences through LSTM.

In MTL-STPN, the training process is that each sub-task individually extracts the features of $f_{tc}^{(l,w)}$ in its target cell through two LSTM layers to obtain feature matrices I_g where g is the label of sub-tasks. The tensor of I_g is $(TS \times N_c)$, where N_c is the number of cells in the previous LSTM layer. Next step, these I_g are concatenated before the sharing layer to obtain a I_{all}, the shape is $TS \times gN_c$, then fed into the sharing layer for capturing the spatial dependency features. Besides, due to the LSTM module, the shared layer enables to model the temporal correlations among the tasks as well. Finally, like STL-STPN, the features obtained by the shared layer are blended with the individual spatiotemporal correlations obtained by the respective dedicated layers, and then train it through LSTM to capture the temporal correlation in the merged feature set and finally predict traffic through the dense network.

Notably, in order to guarantee each task provides an effective contribution to the improvement of the prediction performance, only the top r highest correlated tasks are allowed to be added into the target multi-task model. When r increases, more tasks are involved that ideally the performance will be enhanced while the computational complexity will increase exponentially. On the other hand, if low-relevance tasks are involved, the extracted features may adversely affect the accuracy of the results. Therefore, a correlation measurement algorithm based on spatial-temporal traffic pattern clustering needs to be proposed, which helps to select regions with useful information features for the whole model.

4.3.3 Spatial-Temporal Pearson traffic Pattern Clustering (STPPC)

In MTL-STPN, since the shared layer is LSTM based network, measuring the traffic temporal similarity in the target cell by the Pearson coefficient has become the prior concern. Meanwhile, according to the conclusions drawn from the previous observation in section 5.2, the correlation between the traffic of the cells does not mainly depend on the distance, hence the scope of potential high relevant cell selection extends to the entire map. On the other hand, due to the influence of the dedicated layer and the fusion layer in each task, the target region composed of adjacent cells needs to be considered as well. The reason is that when the traffic in the target cell has high similarity to that in the reference cell, but the target region and reference region has low correlation, the spatiotemporal feature extracted in the MTL model, which is mainly done by the dedicated layer (3D-CNN), will weaken effectiveness of the temporal feature capturing through the shared layer. Therefore, this problem can be formulated as to calculate the matrix similarity Sim$(F_{n_{near}}^{(l,w)}\ F_{n_{near}}^{(l',w')})$, which needs to be considered both spatial and temporal dependencies.

First, the traffic of target cell (i, j) is used to calculate Pearson coefficient with all other cells' flows except the target region cells'. If the value is lower than a threshold z_a, the contrast cell will not be considered in later operating process. The coefficients of remaining cells will be divided into z_v levels with the same interval value and save the level label $z_{(i*,j*)}^{(i,j)}$ from one (highest) to z_v (lowest), where (i*, j*) is the coordinate of contrast cell.

Then, a collection L_{reg} with all the cell ids of the target region needs to be created. Thus the target traffic $f_{rc}^{(i,j)}$ can be found at the centre point (target cell(i,j)) as a typical traffic, and then use this traffic in turn with the traffic $f_{rc}^{(i*,j*)}$ of other cells (i*, j*) in the target region $F_{n_{near}}^{(i,j)}$ for Pearson correlation measurement and record the value $\rho_1^{(i*,j*)}$. $\rho_l^{(i*,j*)}$ is the Pearson coefficient value between typical traffic and contrast traffic, which l starts from one.

The typical pattern cell id will be collected in a set P. When the correlation coefficient is higher than a threshold z_ρ, the contrast cell is labelled B_1 as the same pattern with the target cell (i,j). B_k is label of positive integers which starts from one. Meanwhile, this cell's ID will be removed from the initial set L_{reg}. When all the cell have been calculated, the cell with the lowest correlation coefficient ρ will be the new typical pattern cell. The traffic of this new typical cell will be applied in the Pearson coefficient calculation with the unlabelled cells' flows, and put the new typical cell's ID in the set P. The same process as previous round, the contract cell that $\rho_2^{(i*,j*)}$ is higher than z_ρ will be labelled as B_2, and then remove the ID from L_{reg}. The remaining unlabelled cells will keep updating the $\rho_l^{(i*,j*)}$ once the new value higher than old one. The algorithm will keep operating until the number of unlabelled cell is lower than a threshold z_u, and then marking these unlabelled cells as B_0.

Then the cells in P are the typical cells that have typical traffic pattern. These patterns will be measured Pearson correlation with all other cells except target region cells in the same order and process. The label B_k will be the same as each typical pattern labelling the target region cells. And then the remaining unlabelled cells will be marked as B_0.

At last, the target region with labels B_k will be element-wise compared to the other same size region, to find the same patterns counted as w_l. Then the similarity $S_{(i,j)}^{(i*,j*)}$ can be found through the function $S_{(i,j)}^{(i*,j*)} = \frac{w_l}{n_{near} \times n_{near}}$. The similarity is ordered by $z_s^{(i,j)}$ first, and ordered by $S_{(i,j)}^{(i*,j*)}$ in the same level of $z_{(i*,j*)}^{(i,j)}$.

There are three special cases that need to be considered. First is the importance of z_a, which is used to prevent the possible set with high $S_{(i,j)}^{(i*,j*)}$ but low centre traffic similarity. It will influence the performance of LSTM module. The second is to transformation the Pearson coefficient as absolute value. Due to the characteristics of the Pearson coefficient, only when $\rho = 0$ means the low correlation, otherwise is the positive or negative correlation. The opposite correlation can be modelled through sequence learning algorithms. The third is that when $z_s^{(i,j)}$ and $S_{(i,j)}^{(i*,j*)}$ are the same, the ordering region needs to has similar number of labels as the number of labels in the target region. It is to ensure the completeness and similarity of spatiotemporal feature extraction as much as possible.

When cell (i, j) is used as a target cell for traffic prediction, STPPC can find the previous N_{sim} similar area as sub-tasks in MTL-STPN for traffic prediction.

As a traffic spatiotemporal correlation algorithm suitable for multiple cells, STPPC can perform correlation calculations for target cell and target region traffic. By marking and classifying the traffic patterns in the area through the Pearson coefficient, it can find multiple similar traffic pattern areas. This is very helpful for the extraction of spatiotemporal information and takes into account its importance for the purpose of predicting the traffic in the target cell in this study. The required information is the cell id and its the traffic flow volume.

4.4 Experiment and results

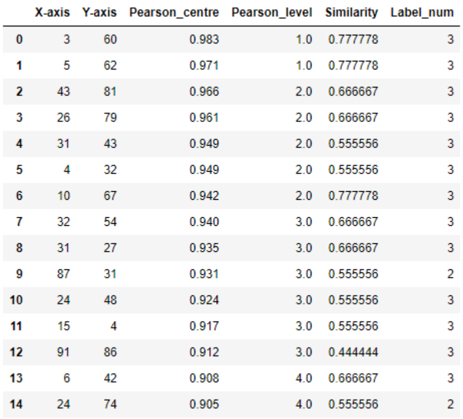
4.4.1 Data processing and hyperparameter configuration

As aforementioned, mobile traffic data in Milan is applied as the model evaluation dataset. The time interval of the data is aggregated from ten minutes to one hour. The geographic range of the dataset is the whole Milan city divided in to 100 X 100 square grid. The time scale is four weeks from November 4th to December 1st, 2013, which has 672 timestamps in total. First two weeks are constructed as training dataset and last two weeks as the testing dataset. To accelerate the convergence process, mobile traffic volume is scaled into range of [0, 1] by Min-Max normalisation method. When predictions are generated, the value will be re-scaled back to its normal scale.

The timestep TS of training dataset is set as 8. n_{near} is set as 3. The input data for 3D-CNN is defined as Samples $\times TS \times n_{near} \times n_{near} \times 1$ which is $324 \times 8 \times 3 \times 3 \times 1$. The input data for LSTM is set as Samples $\times TS \times 1$which is $324 \times 8 \times 1$.

The hyperparameter configuration in the models is optimised by a random search method[14]. In the proposed STL-STPN, the candidate number of layers in the LSTM and 3D-CNN models are from one to eight. The number of cells in each LSTM layer is chosen from 4,8,16,32,64, and the number of filters in the 3D-CNN layer is 4,8,16,32. The epoch is set as 150. Because of the small n_{near} chosen, the kernel size of 3D-CNN is pre-set as (3,3,3) with zero padding in convolutional operation. The best result generated by LSTM layer = 3, 3D-CNN layer = 3, LSTM cell number = {4, 8, 4, 9, 4}, 3D-CNN filter number = {4,8,1}. The fourth LSTM layer and the last 3D-CNN layer have 9 cells and 1 filter, respectively, which is for reconstructing the data to fit the element blend in the fusion layer. When the MTL-STPN is considered, only the LSTM layer after concatenation operation, which is the third layer, will increase its cell number that depends on the number of tasks. The MLP is 2 layers with the cell number = {8,1}. In the proposed STPPN, the z_a and z_ρ are 0.85, and the z_v is 5 that depends on the number of candidate sub-tasks.

4.4.2 Spatial-Temporal Pearson traffic Pattern Clustering test



	X-axis	Y-axis	Pearson_centre	Pearson_level	Similarity	Label_num
0	3	60	0.983	1.0	0.777778	3
1	5	62	0.971	1.0	0.777778	3
2	43	81	0.966	2.0	0.666667	3
3	26	79	0.961	2.0	0.666667	3
4	31	43	0.949	2.0	0.555556	3
5	4	32	0.949	2.0	0.555556	3
6	10	67	0.942	2.0	0.777778	3
7	32	54	0.940	3.0	0.666667	3
8	31	27	0.935	3.0	0.666667	3
9	87	31	0.931	3.0	0.555556	2
10	24	48	0.924	3.0	0.555556	3
11	15	4	0.917	3.0	0.555556	3
12	91	86	0.912	3.0	0.444444	3
13	6	42	0.908	4.0	0.666667	3
14	24	74	0.905	4.0	0.555556	2

Fig. 4.6 The order of similarity generated by STPPN, the target cell is (29,71)

In this research, a task is defined as the short-term mobile traffic prediction for one target region. Because the region contains n_{near} $\times$n$_{near}$ cells. Therefore, the whole city of Milan composed of 100 $\times$ 100 grids can be divided into $100 - n_{near} + 1 \times 100 - n_{near} + 1$, that is, 98 $\times$ 98 reference regions. To locate the region with most effectiveness features extraction for each prediction task, the STPPC algorithm is applied to measure the spatiotemporal correlation between the target region matrix $F_{n_{near}}^{(i,j)}$ and all other reference region matrix $F_{n_{near}}^{(i',j')}$. Then according to the Pearson correlation, spatial pattern and the number of label between target cell traffic and reference cell traffic are sorted. For example, if the location of the target cell is set to (29, 71), after STPPC, the region (3,60) can be found in Fig.4.6

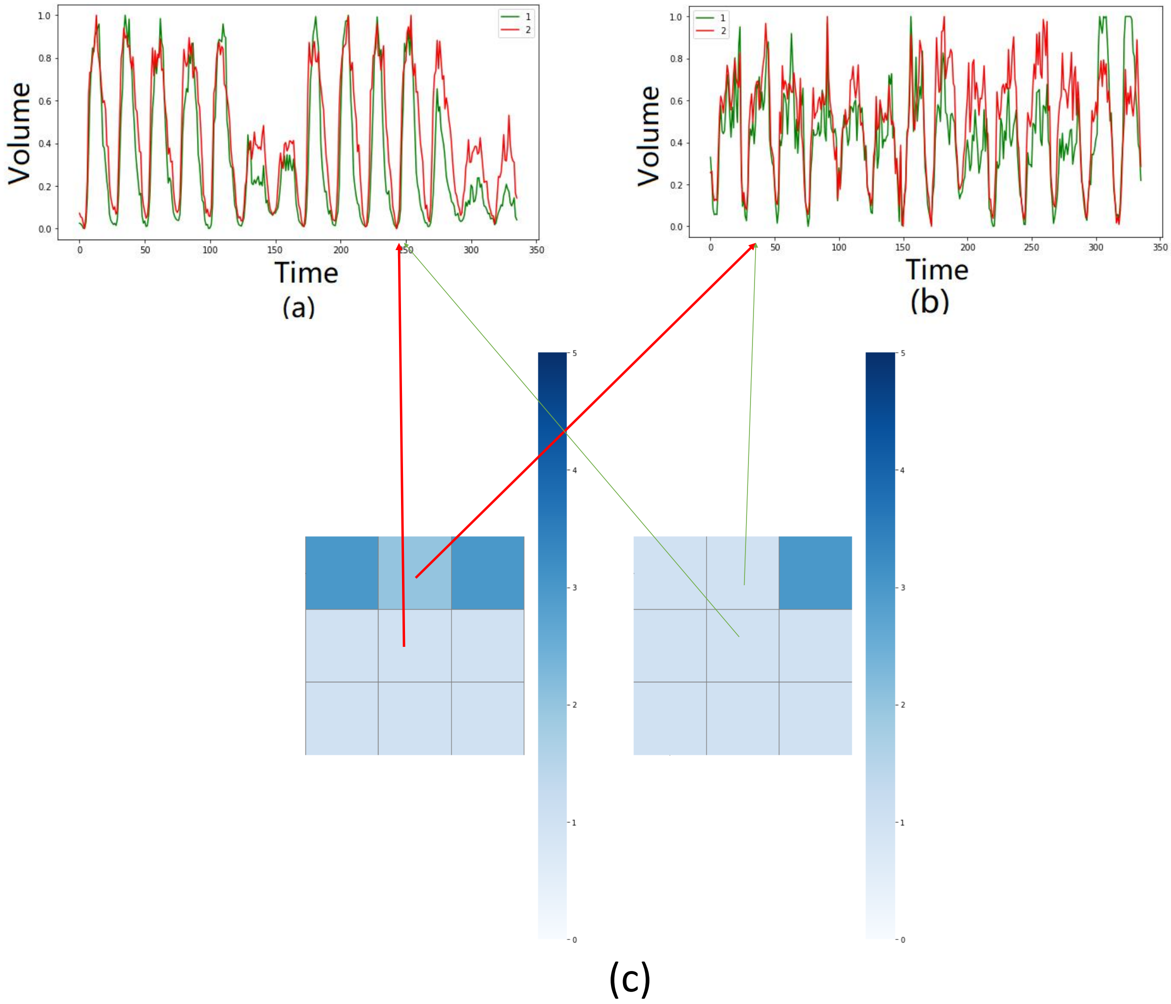


Fig. 4.7 (a) Traffic comparison between cell (29,71) and (3,60), (b) Traffic comparison between cell (29,72) and (3,61), (c) 3 × 3 region centred on (29,71) and (3,60), selected by STPPC and the correlation between the traffic in central grid and all other 8 traffics

according to the rank. When selecting the closest point (3,60) as the reference cell and region, it can be seen the target cell traffic from Fig.4.7, and the labelled pattern distribution in this area, which both show a strong correlation.

By comparing the traffic in each grid in the region, the correlation of the temporal features of several special traffic patterns in the region can be obtained. Then according to the comparison of matrix elements, it can find the spatial dependencies, so as to get the spatiotemporal correlation between the matrices.

4.4.3 Baseline models comparisons

In order to depict the performance difference of the proposed MTL-STPN with the STPPC algorithm as the task filter, various popular machine learning models are applied and compared in STL and MTL scenarios. Timestep is set as 8, which is the same as it used in proposed model, in those baseline models that requires observation window. n_{near} is set as 3×3 for the spatiotemporal feature extraction models.

- LR: Linear Regression is a classic supervised machine learning algorithm that predicts the value through trained function.

- SVR: Support Vector Regression is widely used for regression. By seeking to minimise the structured risk, it can enhance the generalisation ability and fit non-linearity requirement. Besides, SVM can solve high dimension feature problems by the convex quadratic programming.

- STL-LSTM: The structure is the similar to STL-STPN without 3D-CNN layers, which is a typical stack LSTM networks. It has a single input and output. The configuration of the hyperparameters is the same as that in STL-STPN.

- STL-3D-CNN: The structure is the similar to STL-STPN without LSTM layers.

- STL-ConvLSTM: It replaces the LSTM layers to ConvLSTM layers compared to STL-LSTM model. Due to its kernel unit, it can capture long-term temporal feature and spatial dependencies.

In the MTL scenario, compared to the target region (29,71), top 16 highest correlation regions are selected through STPPC as sub-tasks for MTL model. All of the MTL models are fed these 16 traffic matrices, or the sequences as multitask simultaneously to predict the next timestamp traffic in these cells, respectively. These 16 traffic matrices and the temporal sequences are applied to STL models in turn to obtain the prediction value as well.

- MTL-LSTM: The 3D-CNN layers are replaced with LSTM layers compared to MTL-STPN model. Hence both share layer and dedicated layer are LSTM model.

- MTL-3D-CNN: The LSTM layers are replaced with LSTM layers compared to MTL-STPN model.

- MTL-ConvLSTM-3DCNN: The LSTM layers are replaced with ConvLSTM layers compared to MTL-STPN model.

In addition to illustrating the rationality of the proposed framework by modifying the learners or structures, this chapter also selected three state-of-the-art deep learning algorithms. These three deep learning algorithms are specialised in MTLconvolutional networks, and temporal sequence networks, respectively. By utilising these algorithms, it aims to demonstrate the high-performance capabilities of the proposed framework on real-world data.

- MTL-TCNN [177]. This is a deep learning approach that utilises convolutional residual networks for time series analysis. It employs dilated causal convolutions to explore temporal features within sequential data. To prevent model degradation due to increasing layer depth, they apply residual networks to stabilise model optimisation. Additionally, this research includes basic detection of surrounding traffic, based on the assumption that "near things are more related than distant things" [58]. However, this approach does not break away from the limitation of utilising nearby traffic for multi-task learning.

- DC-CNN [138]. This is a multi-layer CNN prediction model that focuses on deep exploration of temporal features. Unlike MTL-TCNN, this research chooses to delve into the temporal aspect. It utilises parallel multiple CNN dense connection networks to model traffic in adjacent time periods simultaneously. Then, through a fusion layer, the features are integrated. This approach effectively utilises CNN models to capture spatiotemporal dependencies within sequential data.

- STCNet [144]. This model utilises ConvLSTM to model correlated data within a region and then employs a point addition method for feature fusion. It is a novel approach that incorporates both deep spatiotemporal exploration through ConvLSTM learners and the perception of multidimensional data. In this chapter, the multidimensional data includes calling traffic and SMS records.

4.4.4 Experimental settings and performance metrics

All the numerical experiments of the proposed model and baseline models are trained on the desktop equipped with i7 6700k CPU, 32 GB memory, and a GeForce GTX 1080 graphic card. The experiments are implemented by python (version 3.6.3) on Windows 10 operating system. Scikit-learn, Tensorflow-GPU[185], and Keras[186] are applied in the python libraries.

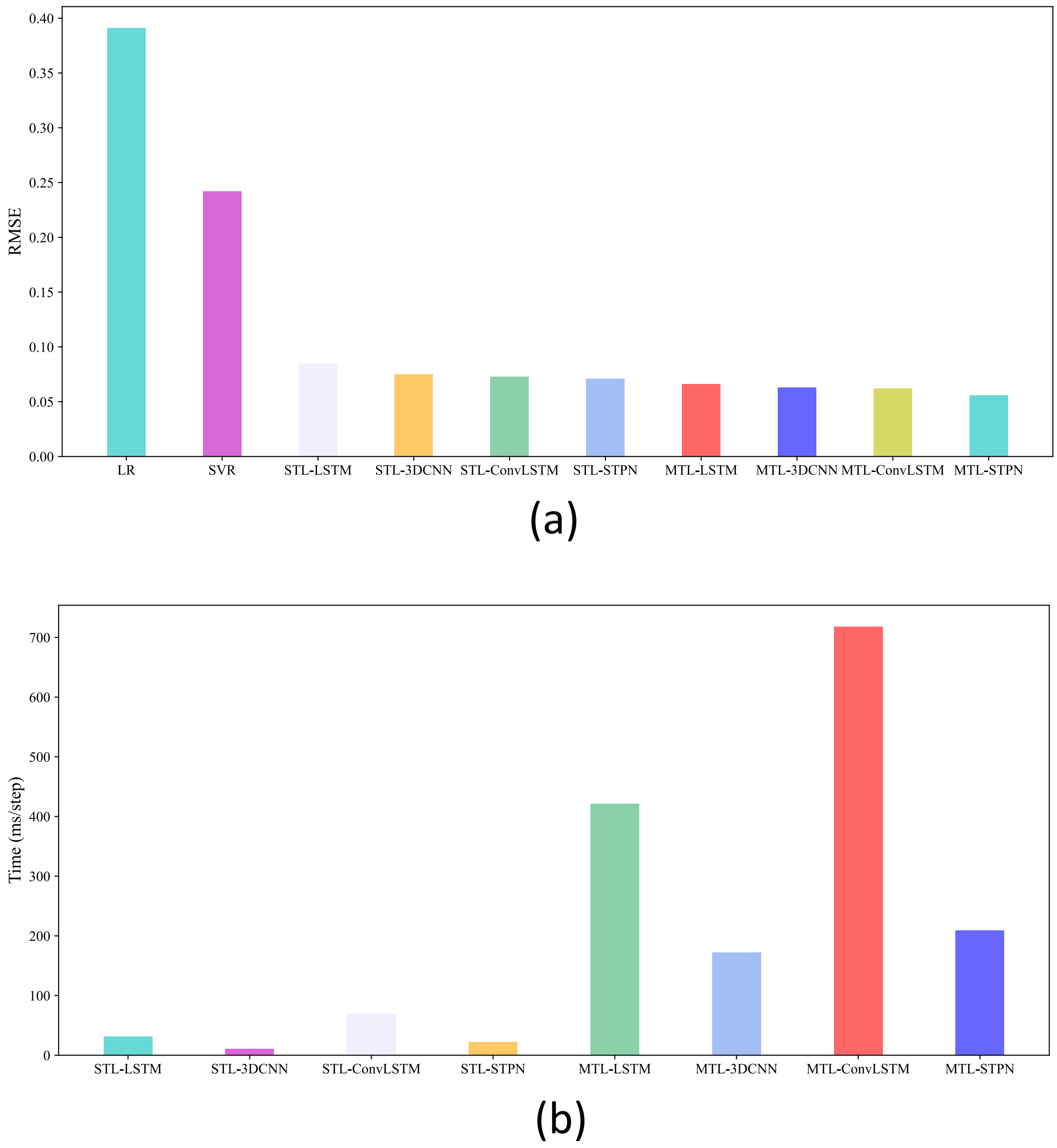
Root mean squared error (RMSE) is applied to evaluate the model performance:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2} \tag{4.1}$$

Where $\hat{Y}_i$ and Y_i are prediction value and ground truth value, respectively. RMSE is used to measure the deviation between the predicted value of the model and the ground truth, which expect the smaller result.

4.4.5 Performance of Proposed Model and the Baseline Methods

Fig.4.8 illustrates the performance of the models over 16 regions in terms of RMSE and the training time consumption. The mean RMSE records the average value of them in 16 regions. Time refers to the time required for training each batch in each epoch. Since LR and SVR do not have such records. However, from the results of the SLIM-TP in the previous chapter and



(a)

(b)

Fig. 4.8 (a)The performance of the similar structure baseline algorithms and proposed model, (b) The training time of the the similar structure baseline algorithms and proposed model

the experience of reviewing other experiments, their operating time is much lower, almost negligible, than these deep learning based models. The target cell is (29,71), and all results are the mean value of 10 repeating experiments.

Firstly, MTL-STPN has the best performance in terms of mean RMSE in these methods, where RMSE(mean) = 0.052. The second best performance is MTL-ConvLSTM, and the main reason of being second is due to the complexity, which is revealed from the training time. MTL-3DCNN also performs well, with being not much different from the results of MTL-ConvLSTM, but the characteristic of the model structure makes it as the least time-consuming model among these MTL models. MTL-LSTM is not as effective as other MTL models, and STL-LSTM generates similar unsatisfactory result comparing to other STL models. But from the relative performance of MTL-LSTM to STL-LSTM, the increase driven by the MTL structure is the most obvious, reaching 22%.

On the other side, in terms of the STL, STL-STPN performed the best. Apart from the shallow learning model, STL-LSTM has the worst effect, which may be resulted by the fact that it is the only model of temporal feature. In addition, regarding to the performance of STL-3DCNN and STL-ConvLSTM, there is no significant difference between the two. It may be caused by the small spatial range of input data, which leads to reaching the limitation of the convolutional network. The method of fusing the features of the centre point and the nearby area can indeed further improve the effect. Finally, from the overall point of view, MTL-based models are generally performed better than the STL-based models, but the downside is that the complexity is also drastically increased. However, due to the shared layer and parallel learning structure, the unit time of the MTL-based model to run once is less than the time required to repeat the STL-based model fifteen times.

In summary, the structure of share-dedicated layer in proposed MTL model can indeed significantly improve the performance of the proposed STL model. The reason is that the shared layer is a temporal feature model, and the features of spatiotemporal can be further captured through MTL to improve the performance. However, if the model only capture the spatiotemporal feature, the improvement is not obvious, which has been shown in MTL-3DCNN and MTL-ConvLSTM results. On the other hand, the MTL-LSTM have a impressed improvement in the performance compared with STL-LSTM. Therefore, the structure of temporal, spatial-temporal, and fusion layer can provide good prediction performance while being limited in complexity increasing.

Fig.4.9 illustrates the performance of the state-of-the-art models in terms of RMSE and the training time consumption. In the comparison of state-of-the-art methods, DC-CNN has the highest RMSE because it lacks sufficient exploration of temporal features. It lacks an

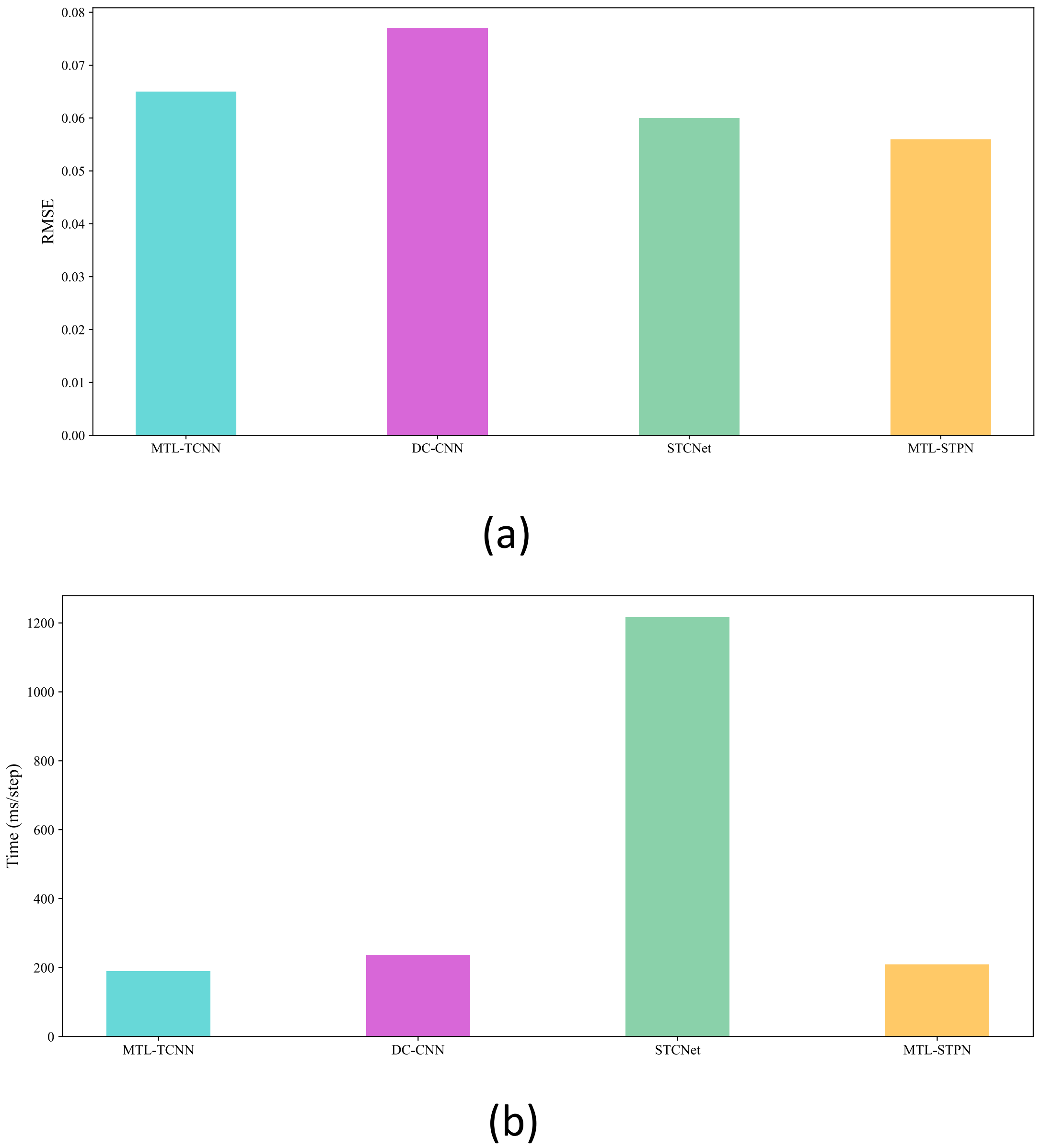


Fig. 4.9 (a)The performance of the state-of-the-art baseline algorithms and proposed model, (b) The training time of the the state-of-the-art baseline algorithms and proposed model

effective module to prevent model degradation, resulting in insufficient depth of the CNN layers. For time sequence analysis, preserving a sufficient length of historical information is crucial. On the other hand, MTL-TCNN improves the CNN's sensitivity to temporal features through its own residual network and dilated convolutions. However, due to the idea that proximity implies higher correlation, it cannot discover more effective traffic information.

This limitation becomes more pronounced when there are significant differences between the target area and nearby traffic, leading to a decrease in accuracy.STCNet, based on ConvLSTM, does exhibit similar accuracy to the proposed model, but it lacks an effective spatial filtering mechanism and suffers from long training times, which are still shortcomings.

MTL-STPN outperforms the baseline methods. On one hand, MTL-STPN employs parallel LSTM, 3D-CNN networks, and feature fusion to enhance the extraction of consecutive spatiotemporal features. On the other hand, MTL-STPN incorporates the STPPC algorithm, which enhances the ability to extract dependencies between network traffic features in other grid cells, thereby further improving prediction performance.

4.4.6 The performance of MTL-STPN and other MTL-based models

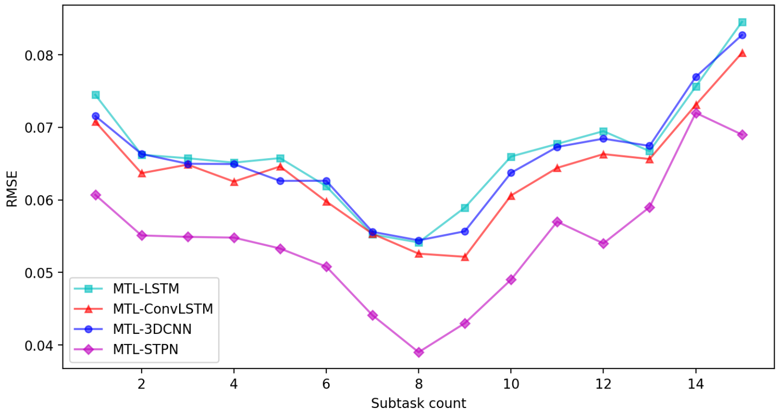


Fig. 4.10 Performance of four MTL-based model

Fig.4.10 illustrates the traffic prediction performance for the target cell (29, 71) after the 16 regions, which have the largest correlation with the target cell being found by STPPC sorting, being added to the four models of MTL in turn. First of all, when the first 8 tasks being added in sequence, they have shown a similar trend of predicting better performance as tasks increase, which demonstrates that with the increase of sub-tasks, the performance of each model can be improved, and the MTL-STPN model has the best performance. Then after the seventh or eighth tasks, all models are negatively affected by the new tasks, which is likely caused by two reasons. The first reason is that the features of spatiotemporal are already redundant in the task. Not only can it not make a positive contribution to the whole system, but on the contrary it will 'dilute' the more critical features, and consequently making the performance worse. And in order to control variables, their number of layers and kernel size are kept the same as MTL-STPN, therefore, the critical point is also similar to MTL-STPN. The second reason might be due to the affected data, because it is through sorting that we find the most relevant area, the model performance declines due to weak spatiotemporal dependencies in the subsequent data. This issue will be discussed in details in the next subsection

Another phenomenon worth discussing is that after the start of the eighth task, the differences between the three models were not significantly different from each other begin to manifest, especially for the MTL-ConvLSTM. Although the performance of MTL-ConvLSTM is also declining, it is obviously more stable. From the result of the last 15th point, we can also see that the two models with the convolutional layers performed slightly better than the MTL-LSTM model which does not have convolutional layers. It can be interpreted as that the feature extraction of spatiotemporal does require the addition of a convolutional layer, and it cannot be compensated by structural changes (MTL, Stack learning) alone, which is another reason of why the dedicated layer must be added to the convolutional layer in MTL-STPN..

4.4.7 The dependencies of STPPC selected data

According to the conclusion in the previous section, which says that the performance of the model decreases may be due to the decreasing in data relevance,16 regions are inverted. Besides, the worst 16 regions searched through the full map are fed into the MTL-STPN model to check whether the data selected by STPPC is helpful for the feature extraction of the model.

First of all, It can be seen that the results in the previous section are marked as 'STPPC top 16'. As a comparison item, the 16 sub-tasks are reversed and added to the model in turn. It has shown that the general trend of the data remains relevant to the comparison item, but in the tasks such as the seventh or eighty-ninth task, the performance is weak with the result not

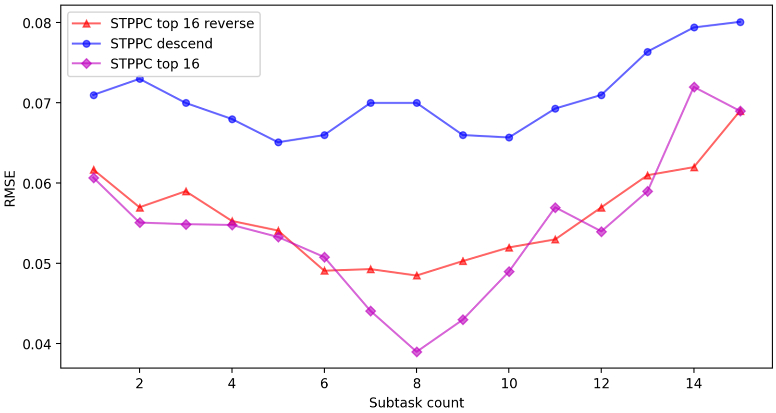


Fig. 4.11 Performance of STPPC algorithm in proposed MTL-STPN

being to the ‘bottom’. After that, the RMSE gradually increased. This pattern shows that the trend is in line with the conclusion in the previous section, which means that the trend rebound may be due to the feature redundancy. However, the result of 7, 8, and 9 sub-tasks do not reach the best results, which proves that the correlation is related to feature extraction. It is the spatiotemporal features extracted due to the similar characteristics of the first eight tasks that can achieve the best results. However, the features of the last eight data (that is, the first eight data of ‘STPPC top 16 reverse’) cannot achieve this effect.

In addition, it has been selected the sixteen most irrelevant regions and put them into the model, and found that the overall effect was not good, and the effect was worsened with the increase of subsequent tasks, which once again verified the existence of feature redundancy. From here, it can be seen that the region selected by STPPC does help the current MTL-STPN model to extract key features, thereby improving the effect.

4.5 Conclusion

In this chapter, a Multi-Task Learning Spatial-Temporal Parallel deep learning Network is proposed for mobile traffic prediction. Meanwhile, a model based on the parallel structure of LSTM, 3D-CNN and fusion layer is designed as the basic model for each task in MTL. In addition, this study also designed a regional temporal and spatial correlation detection algorithm STPPC to select cooperative tasks with effective features for MTL. It has demonstrated the effectiveness of the proposed model and algorithm through the Milan open dataset, and found that MTL-STPN has better prediction accuracy while maintaining a relatively low system complexity and shorter training time by comparing the performance of the traditional machine learning algorithms and the latest deep learning based model. Moreover, the structure of MTL-STPN can indeed discover more spatiotemporal characteristics by increasing the number of tasks within a certain range, thereby enhancing performance. Lastly, STPPC has also been proven to be an effective algorithm to find the area with the strongest correlation with the target cell and provide effective spatiotemporal dependencies for the proposed mode.

Chapter 5

SLIM-TP: a deep Stack Learning-based framework for Instant Message Traffic Prediction

In the previous chapter, efficient prediction of small-scale base station-level network traffic was achieved by deep exploration of the spatiotemporal dependencies within the base station-level traffic information. Therefore, in complex scenarios, feature extraction from the data is crucial. In the face of applications with stronger burstiness in network traffic, new highly correlated data needs to be identified to improve prediction performance. Building upon the user-population-based feature extraction and the exploration of user periodic behaviour in Chapter 3, this chapter proposes a deep stack learning-based framework for instant message traffic prediction (SLIM-TP), based on the excellent architecture of MTL from the previous chapter. SLIM-TP leverages the advantages of the framework to effectively explore the spatiotemporal correlations in traffic and the spatiotemporal dependencies between traffic and population. Moreover, it innovatively captures the characteristics of these explored features through a meta-learner. After conducting global auxiliary information feature modelling, SLIM-TP improves instant messageIM traffic prediction by 30

5.1 Introduction

] Over the last decade, the dramatic development of cellular networks has revolutionised the way of user communication. IM application, such as WeChat, Whatsapp, and Facebook Messenger, has been installed in 89 percent of mobile devices and cost roughly half of the total mobile apps operating hour[168]. Moreover, in the report from WalktheChat in 2018[169], WeChat (the most popular IM app in China) accounts for 34 percent of total mobile data traffic in China. To reduce the OPEX of application providers, meet the mobile users' QoS, or even understand the social behaviours of human beings, traffic prediction for IM applications, especially for WeChat, will be of great importance in 5G and beyond mobile networks.

As a classical research topic, mobile network traffic prediction has attracted a lot of research interest from both academia and industry. Generally, the proposed prediction methods can be divided into two categories: model-driven methods and machine learning-based methods. For the first category of prediction methods, the time series of mobile traffic are fitted to specific mathematical models and the future traffic loads are predicted based on statistics or probabilistic distributions. In[129], the linear ARIMA model has been used to capture the short-term correlation in mobile network traffic. As an extension, the seasonal ARIMA model has been adopted to improve the ARIMA model on long-term traffic correlation capturing in[170]. Li *et al.*[131] demonstrated that the mobile traffic loads possess a strong self-similarity and utilised the -stable model to predict the mobile traffic fluctuations. However, it is difficult for these model-driven methods to estimate realistic mobile traffic accurately since mobile traffic is much more complex than mathematical models with various irregular patterns.

For the second category of methods, more and more studies have adopted machine learning technologies for mobile network traffic prediction in recent years. In[135] and[136], the LR and the SVR are proposed to predict the future traffic loads of mobile cells, respectively. Nevertheless, due to the fact that they cannot perform feature extraction on their own while relying on some prior knowledge of the input features, these shallow learning methods cannot cope with many practical prediction scenarios. Powerful deep-learning tools have recently been leveraged for mobile traffic prediction. Qiu *et al.*[81] trained a RNN to predict the cellular level mobile traffic due to its capability of capturing the spatiotemporal correlations in traffic load time series. Assuming traffic information of neighbouring cells, Feng *et al.*[143] proposed an LSTM network-based prediction model to forecast the traffic loads of a target cell. Furthermore, based on historical traffic loads generated in all the cells, a CNN based prediction model[138] and a ConvLSTM network-based prediction model[71] were proposed to forecast the spatial mobile traffic distribution in a city. However, most of the existing deep learning-based methods are proposed for the aggregated mobile traffic generated by all the applications, which vary relatively stable and regularly. Unlike aggregated mobile traffic, the WeChat traffic pattern has obvious burstiness and randomness. Furthermore, the trend and volume of WeChat traffic are influenced seriously by the variation of crowd distribution, which is ignored in most of the existing prediction methods.

Therefore, it is challenging to adopt the existing deep learning-based methods for WeChat traffic prediction directly since they will cause obvious prediction errors. To fill the above gaps, I introduce deep stacking learning architecture into mobile traffic forecasting and propose SLIM-TP, a deep stacking learning-based framework for WeChat (IM) traffic prediction especially. The main contributions of this work are summarised as follows:

- Through the statistical analyses of conditional distribution entropy, this chapter demonstrates that the WeChat traffic pattern possesses stronger burstiness and randomness than the aggregated mobile traffic, and the variation of crowd distribution has an important influence on the WeChat traffic trend;

- Deep stacking learning is introduced into WeChat traffic prediction and proposed as a novel prediction framework, SLIM-TP. In SLIM-TP, three base-learners are designed to forecast the future WeChat traffic load respectively based on the temporal sequence pattern and spatiotemporal correlations of WeChat traffic as well as the correlations between the WeChat traffic and the crowd distribution. Considering the burstiness and randomness of the WeChat traffic pattern, a Gaussian Process(GP)-based referee is introduced to judge the reliability of each base-learner. Finally, SLIM-TP utilises a deep neural network-based meta-learner to automatically extract the features hidden in the three base-learners' outputs as well as their reliability, and generate the ultimate prediction result;

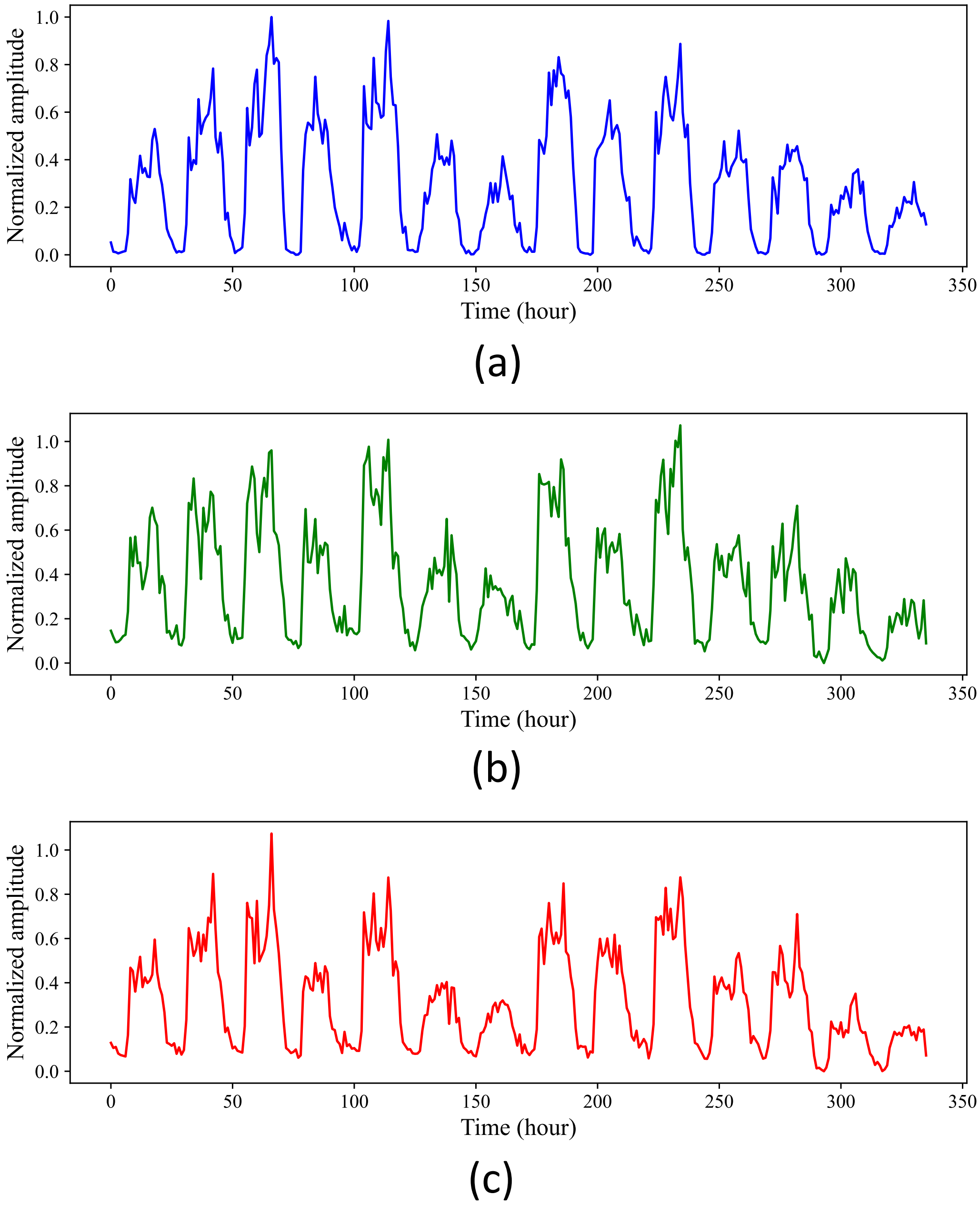
- This chapter also evaluates the performance of SLIM-TP using real-world WeChat traffic traces collected in Guangzhou. I provide insights into how stacking learning can improve the framework's prediction accuracy.
- In this chapter, the model is applied to the real data from the previous chapter, confirming the effectiveness of using multidimensional information as auxiliary data in small-scale network traffic prediction.

The rest of this chapter is organised as follows. Section 5.2 presents the statistical characteristics of WeChat traffic, followed by a brief introduction to stacking learning technology. Section 5.3 elaborates the proposed SLIM-TP. In Section 5.4, it evaluates the performance of SLIM-TP in comparison with some state-of-the-art prediction methods. In Section 5.5, I summarise this study and discuss future work.

5.2 Data and Challenges

5.2.1 Description of the IM traffic trace

In this chapter, it adapts the real-world traffic records of WeChat, the most popular IM application in China, generated in Guangzhou from 03/01/2019 to 03/31/2019 provided by Unicom China. Specifically, the Guangzhou city area is divided into 201×201 grids with the same size of $100m \times 75m$. Each WeChat record contains information about its timestamp, geographical grid, and data volume. It refers to each grid as a cell and regards the forecasting problem of one cell's future WeChat traffic load as a cellular-level IM traffic prediction task. Fig. 1 illustrates the normalized WeChat traffic loads and aggregated mobile network traffic loads in three cells belonging to a commercial district, a residential district, and a market, respectively, during two weeks.



(a)

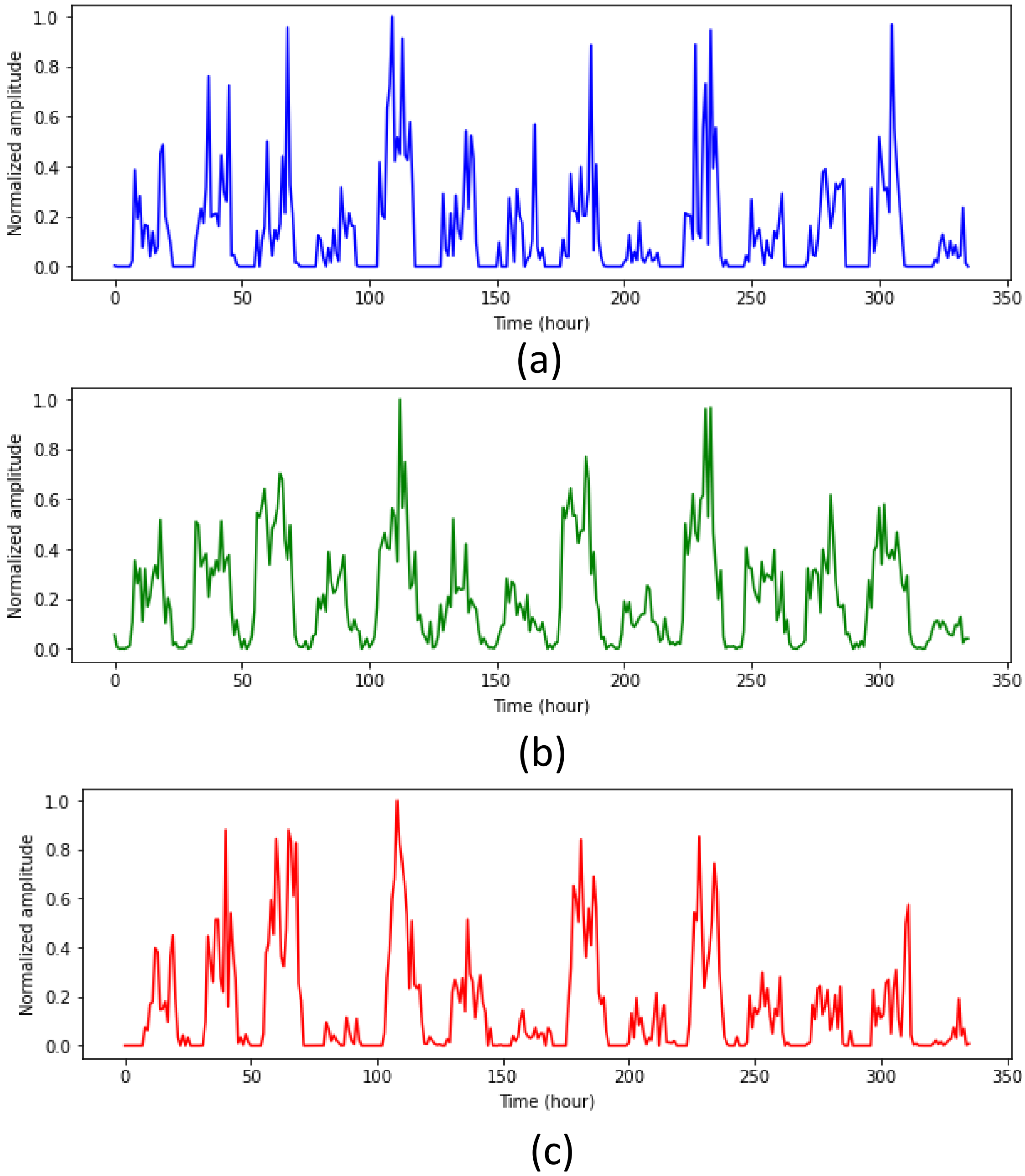
(b)

(c)

Fig. 5.1 The aggregate network traffic at (a)(14,38), (b)(53,34), (c)(94,35)

Specifically, the aggregated mobile traffic load contributed by all the mobile applications and the mobile user number in each cell during every secular hour from 03/01/2019 to 03/31/2019 are also provided by Unicom China as text documents.

This study does not breach user privacy or raise ethical or legal issues. Indeed, It does not need to process individual or personal data. Also, the traffic records are strongly anonymized



(a)

(b)

(c)

Fig. 5.2 The IM traffic at (a)(14,38), (b)(53,34), (c)(94,35)

by the geographical aggregation at the cellular level. All these ensure that the mobile demands are merged over numerous subscribers.

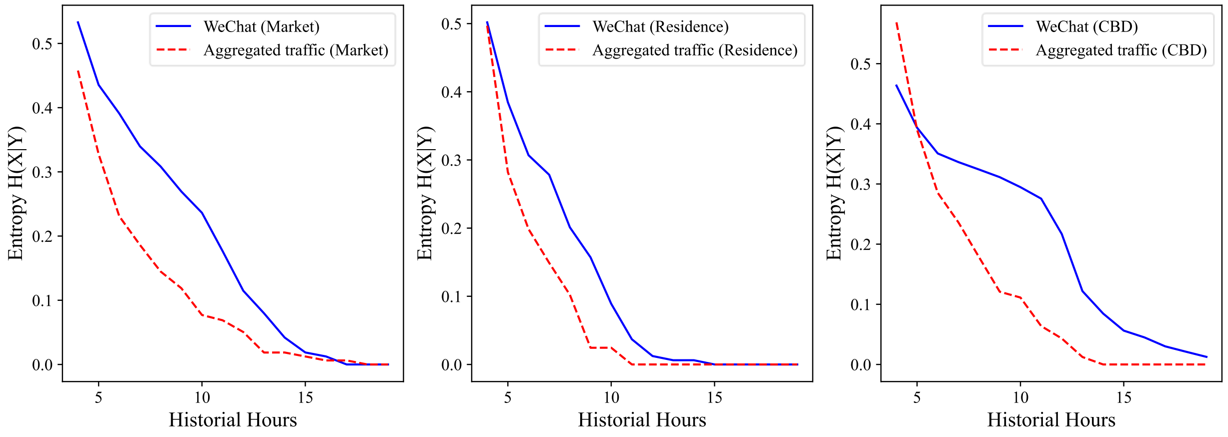


Fig. 5.3 Temporal CE to historical time of WeChat traffic at (a)(14,38), (b)(53,34), (c)(94,35)

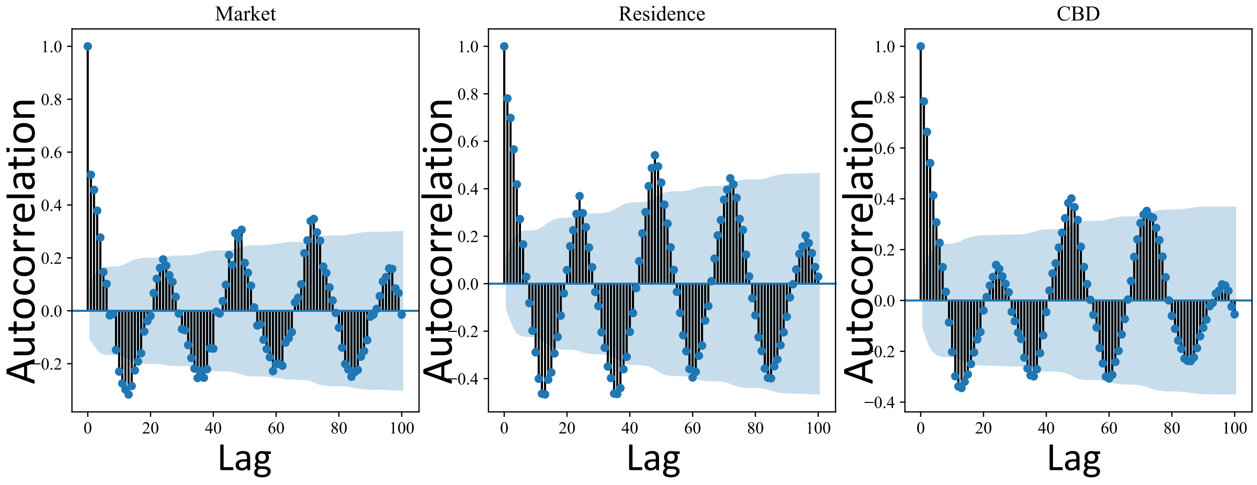


Fig. 5.4 The autocorrelation analysis of WeChat traffic at (a)(14,38), (b)(53,34), (c)(94,35)

5.2.2 Statistical analyses for IM traffic patterns

It sets the time interval resolution at one hour following the settings in [171]. I use $r_i[t]$ to record the WeChat traffic load of cell i generated in the tth time interval. The WeChat traffic loads are divided into ten levels, and Fig.5.3 demonstrates the average CEs [133] of load level distribution at each time interval considering various numbers of former time intervals for

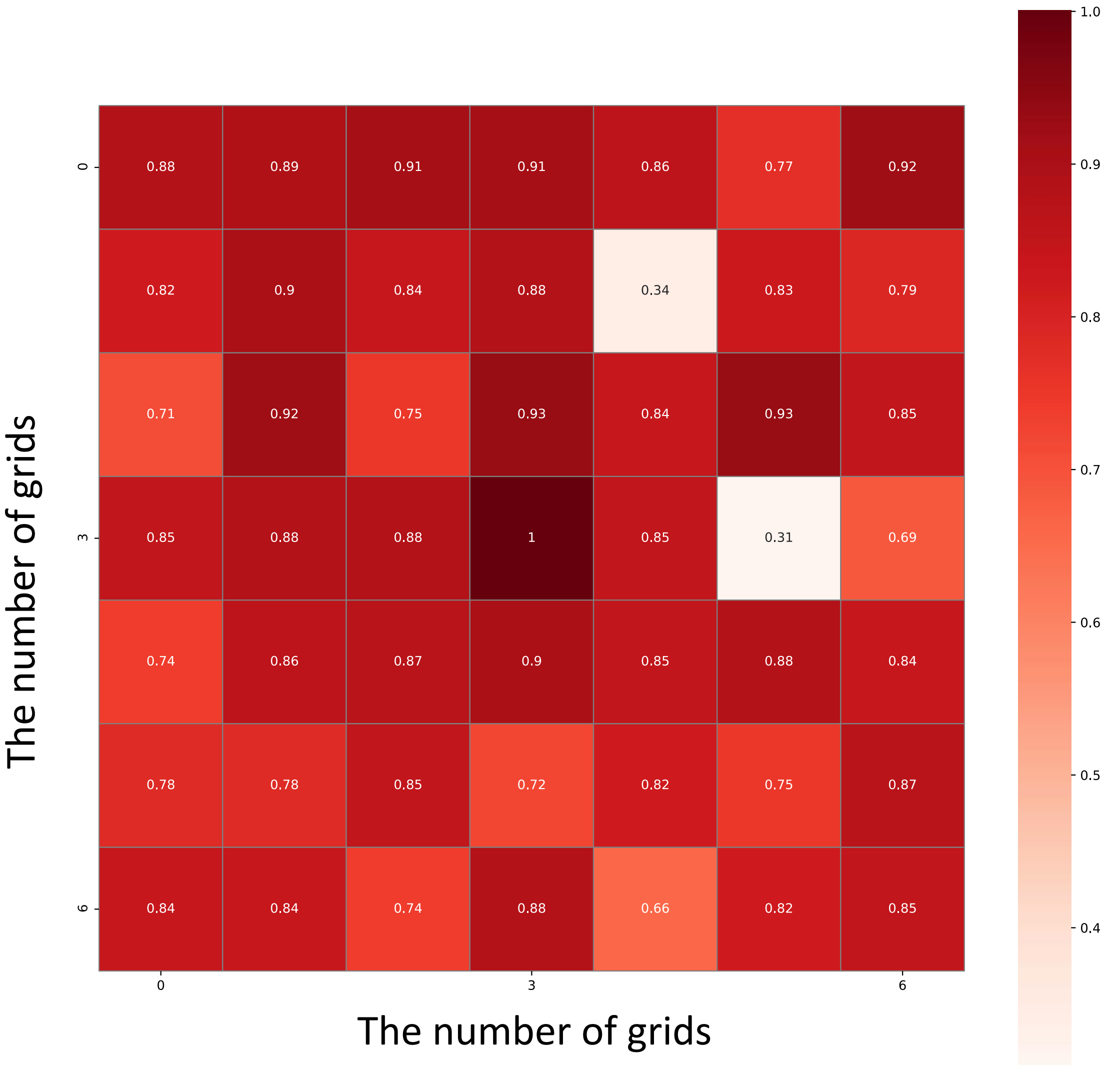


Fig. 5.5 Spatial correlation analysis of WeChat traffic

the WeChat traffic and the aggregated traffic, respectively, in the same three cells in Fig.5.1. CE reflects the randomness of traffic load distribution when the traffic loads in former time intervals are given. From Fig.5.1, Fig.5.2, Fig.5.3, Fig.5.4, Fig.5.5and Fig.5.6, it has the following observations.

Observation 1: The cellular level IM traffic patterns are less regular than the patterns of the aggregated traffic. The randomness of IM traffic load distribution is also much larger than that of the aggregated traffic when I know their former loads, which implies

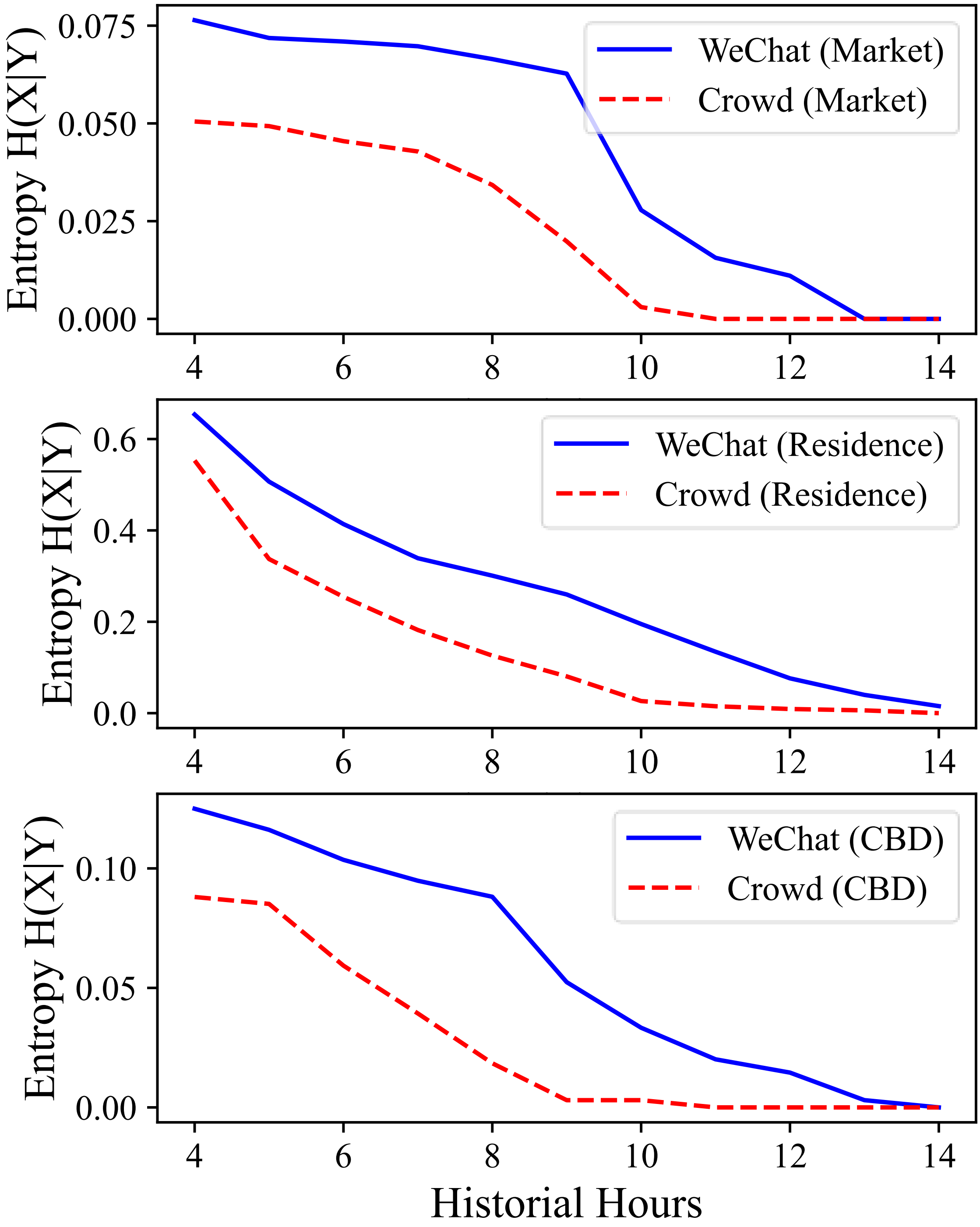


Fig. 5.6 The spatial CE by nine adjacent grids between WeChat and crowd traffic

that forecasting the future load of IM traffic using previous values is harder than that in conventional prediction tasks focusing on the aggregated traffic.

Fig.5.4 displays the autocorrelations of the WeChat traffic in the three cells, while Fig5.5 shows the Pearson correlation coefficients between the market cell (14,38) WeChat traffic load series and the WeChat traffic load series in its neighbour cells.

Observation 2: The cellular level IM traffic exhibits obvious autocorrelations in time domain and the spatial correlations indeed exist among the IM traffic patterns generated in different cells. This indicates that the future IM traffic load in a certain cell may be predicted through the historical IM traffic loads of this cell and the neighbour cells.

By dividing the traffic loads into ten levels, it can get more detailed information on the WeChat load distribution (e.g. measures by CE). Fig.5.6 demonstrates CE values of WeChat load distribution at cells (14,38), (53,34), and (94,35), respectively, when I consider the historical mobile user distributions or the WeChat traffic loads in each cell's neighbour cells during previous time intervals. From Fig5.6, it can clearly observe the following conclusion.

Observation 3: Compared with the historical IM traffic loads in a cell's neighbour cells, the mobile user distributions within the cell's neighbour area in a few of previous time intervals seem to have a more important influence on this cell's future IM traffic load.

Due to the high randomness of IM traffic loads and the strong correlation between the IM traffic patterns and mobile user distributions, the existing prediction methods which are mainly developed for the aggregated mobile network traffic are not proper in IM traffic forecasting. Therefore, it is necessary to design a traffic prediction method, especially for IM applications that can provide highly reliable prediction results based on the autocorrelations and spatiotemporal correlations of IM traffic patterns as well as the correlations between IM traffic variation and the mobile user distributions in previous time intervals.

5.3 The Proposed Framework for IM Traffic Prediction, SLIM-TP

Fig.5.7 shows the structure of the proposed SLIM-TP, which is a deep stack learning based framework for cellular-level IM traffic prediction. The SLIM-TP consists of five components: three heterogeneous base-learners that give the unreliable prediction values of the IM traffic

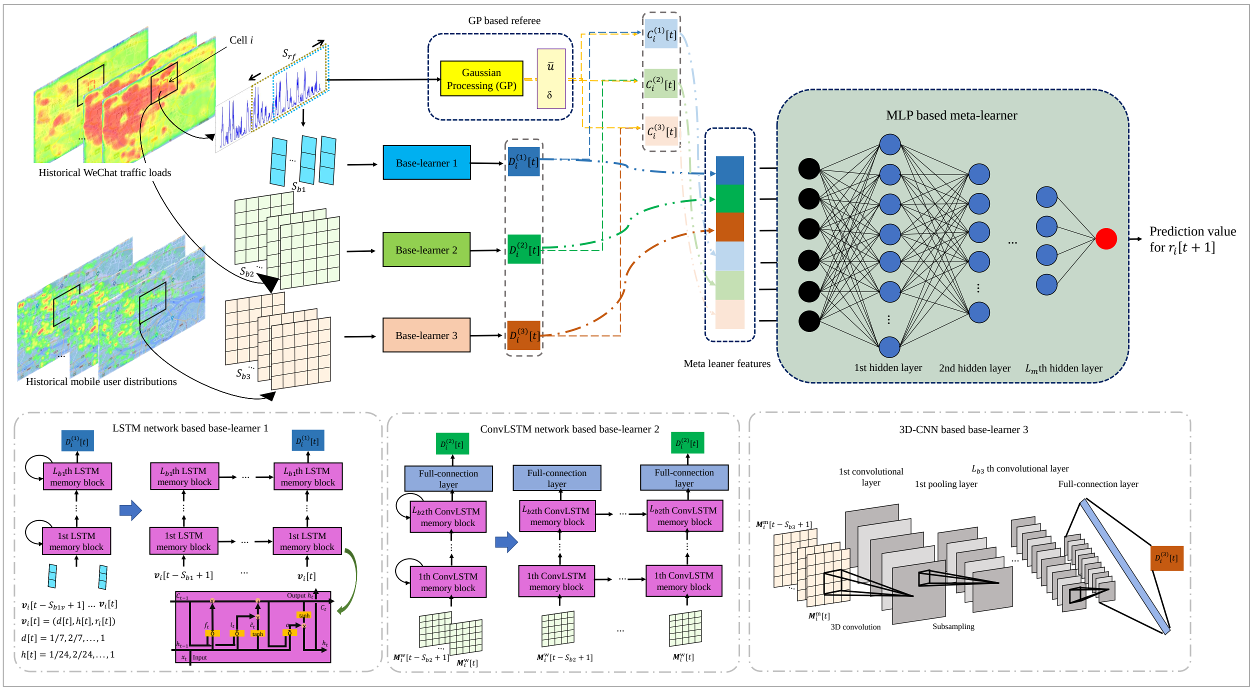


Fig. 5.7 The architecture of SLIM-TP and the structure of three base-learners

load in the next time interval, a referee that judges the reliability of each base-learner as well as generates the meta-features for the meta-learner, and a meta-learner that provides the final reliable prediction result based on the input meta-features. The details of the proposed framework are introduced as follows.

5.3.1 The Deep Learning Based Base-learners

As presented in Fig.5.7, there are three base-learners in SLIM-TP to roughly predict the IM traffic load of a certain cell in the next time interval. In order to fully utilise the statistical properties of the IM traffic observed above, such as autocorrelations, spatiotemporal correlations, and correlations with the historical mobile user distributions, the three base-learners are designed based on LSTM network, ConvLSTM network, and 3D-CNN, respectively.

LSTM network-based base-learner 1 :

In SLIM-TP, it constructs base-learner 1 as a multi-layer LSTM network with L_{b1} layers of LSTM memory blocks to predict the future IM traffic load in a target cell using this cell's historical IM traffic load records. Each LSTM memory block is logically a recurrently connected subnet containing some functional modules called gates. According to their corresponding practical functionalities, these gates are classified as input gates, input activation gate, forget gate and output gate. With the LSTM network's ability to model the long-term dependency in time series, base-learner 1 is designed to capture the autocorrelations of IM traffic patterns.

For a target cell i and a certain time interval t, base-learner 1 will be continuously fed a sequence of input vectors related to previous S_{b1} time intervals, $(\mathbf{v}_i[t-S_{b1}+1],...,\mathbf{v}_i[t])$, and predict (as the output) the IM traffic load in next time interval $t+1$. Each input vector $\mathbf{v}_i[t]$ consists of three attributes: one is the IM traffic load in this time interval, $r_i[t]$, and the other two attributions, $d[t]$ and $h[t]$, reflect t's temporal information (which day in a week and which hour in a day).

ConvLSTM network-based base-learner 2 :

Base-learner 2 in SLIM-TP is designed as a ConvLSTM network-based prediction model to extract the spatiotemporal correlations of IM traffic loads generated in the target cell and the adjacent cells.

As can be seen from Fig.5.7, base-learner 2 consists of L_{b2} layers of ConvLSTM memory blocks, followed by a full-connection layer of neural network. The input of base-learner 2 is a sequence of S_{b2} matrices, $(\mathbf{M}_i^w[t-S_{b2}+1],...,\mathbf{M}_i^w[t])$. Each matrix has the size of $n\times n$ and records the IM traffic loads in cell i's $n\times n$ adjacent cells in a previous time interval. The output of base-learner 2 is the predicted value for the target cell's IM traffic load in the next time interval $t+1$.

3D-CNN based base-learner 3 :

It constructs base-learner 3 as a 3D-CNN based prediction model to capture the correlations between a target cell's future IM traffic load and the mobile user distributions within this cell's neighbour area in a few previous time intervals.

The input of base-learner 3 for mobile cell i can be seen as video-like data with S_{b3} matrices, $(\mathbf{M}_i^m[t-S_{b3}+1],...,\mathbf{M}_i^m[t])$. These matrices are with the same size of $n\times n$ and record the numbers of mobile users distributed in each of cell i's $n \times n$ adjacent cells in previous S_{b3} time intervals. As shown in Fig.5.7, base-learner 3 comprises L_{b3} convolutional layers and L_{b3} pooling layers, and the outputs of the last pooling layer are converted into the final output value representing the predicted IM traffic load of cell i in time interval $t+1$ by a full-connection layer of the neural network.

5.3.2 The GP Based Referee

Considering Observation 1 that the distribution of IM traffic load has strong randomness, the results of the three base-learners are often unreliable. In SLIM-TP, it uses a GP-based referee to judge the credibility of each base-learner and generate meta-features for the meta-learner. The GP, which is an important class of Bayesian non-parametric machine learning algorithms, encodes domain or expert knowledge into the kernel function and thus has great interpretability for its outputs [172]. The GP-based referee will provide the expectation and variance of the IM traffic load of a target cell in the next time interval instead of a specific prediction value.

Mathematically, the GP-based referee works as follows. Given the IM traffic loads of the target cell i in previous S_{rf} time intervals, $r_i[t-S_{rf}+1]$, $r_i[t-S_{rf}+2]$, ..., and $r_i[t]$, the posterior distribution of this cell's IM traffic load in next time interval, $r_i[t+1]$, can be derived as

$$r_i[t+1] \sim N(\bar{\mu}, \sigma) \tag{5.1}$$

, where $r_i[t+1]$'s expectation and variance are respectively calculated as

$$\bar{\mu} = \mathbf{k}_*^{\mathbf{T}} \cdot \mathbf{K}^{-\mathbf{1}} \cdot [\mathbf{r_i}[\mathbf{t}-\mathbf{S_{rf}}+\mathbf{1}], \mathbf{r_i}[\mathbf{t}-\mathbf{S_{rf}}+\mathbf{2}], \ldots, \mathbf{r_i}[\mathbf{t}]] \tag{5.2}$$

$$\sigma = k_{**} - \mathbf{k}_*^{\mathbf{T}} \cdot \mathbf{K}^{-\mathbf{1}} \cdot \mathbf{K}_* \tag{5.3}$$

$\mathbf{K}$ is an $S_{rf} \times S_{rf}$ kernel matrix, whose element $\mathbf{K}(\mathbf{a}, \mathbf{b})$ equals $k(t-S_{rf}+a, t-S_{rf}+b)$. $\mathbf{k}_*$ is an $S_{rf} \times 1$ kernel vector, whose ath element equals $k(t-S_{rf}+a, t+1)$. k_{**} is a scalar with the value of $k(t+1, t+1)$. Specifically, $k(t_1, t_2)$ is the selected kernel function.

With the output of $D_i^{(j)}[t]$, the credibility of the jth base-learner (j=1, 2, or 3), $C_i^{(j)}[t]$, is calculated as $r_i[t+1]$'s probability density at the value of $D_i^{(j)}[t]$:

$$C_i^{(j)}[t] = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\left(D_i^{(j)}[t]-\bar{\mu}\right)^2}{2\sigma^2}} \tag{5.4}$$

Obviously, the more $D_i^{(j)}[t]$ is close to $r_i[t+1]$'s expectation, $\bar{\mu}$, calculated by the referee, the larger credibility the jth base-learner will have. Combining the three base-learners' prediction results with their corresponding credibilities, the meta-features for mobile cell i in the tth time interval are obtained. These meta-features can be expressed by a vector containing six elements, $(D_i^{(1)}[t], C_i^{(1)}[t], D_i^{(2)}[t], C_i^{(2)}[t], D_i^{(3)}[t], C_i^{(3)}[t])$.

5.3.3 The MLP Based Meta-learner

Intuitively, since all of the three base-learners for mobile cell i try to give the prediction values using only partial characteristics of the IM traffic patterns, their outputs and credibilities in the tth time interval (the meta-features provided by the referee) will have a strong correlation with the ground-true IM traffic load in the next time interval, $r_i[t+1]$. In SLIM-TP, an MLP is used as the meta-learner to non-explicitly extract this correlation and generate the final prediction result according to the meta-features.

As shown in Fig.5.7, the MLP-based meta-learner has L_m hidden layers, where the neurons in each hidden layer are densely connected to the neurons in its former or later layer. The input layer and output layer of the meta-learner have six neurons and one neuron, respectively, representing the vector of meta-features and the final prediction result of the IM traffic load for a target cell. With the MLP's ability of feature extraction and correlation characterisation [173], the meta-learner is expected to further improve the three base-learners' prediction accuracy.

5.3.4 Constructing Training Sets to Train the Base-learners and the Meta-learner in SLIM-TP

As shown in Fig.5.8, it divides the associated records about the WeChat traffic loads and mobile user distributions into three periods, each of which comprises T_1, T_2, or T_3 time intervals, respectively. For an arbitrary target mobile cell i, it constructs the training sets for the three base-learners as well as the meta-learner in SLIM-TP as follows. A sliding window with size S_{b1} is applied to split the WeChat traffic load series of the ith mobile cell in the first period, $\mathbf{r}_i^{T_1} = (r_i[1], r_i[2], \ldots, r_i[T_1])$, and generate the training set of $T_1 - S_{b1}$ base-samples for base-learner 1 by labelling each sequence of S_{b1} input vectors with cell i's WeChat traffic load in next time interval.

This chapter uses a 3D matrix $\mathbf{M}_i^{wT_1} = (M_i^w[1], M_i^w[2], \ldots, M_i^w[T_1])$ with size $n \times n \times T_1$ to record the WeChat traffic load distributions in the area of cell i's $n \times n$ adjacent cells during the first period and apply a 3D sliding window with size $n \times n \times S_{b2}$ to split $\mathbf{M}_i^{wT_1}$. By labelling each $n \times n \times S_{b2}$ matrix with cell i's WeChat traffic load in the next time interval, it generates the training set of $T_1 - S_{b2}$ base-samples for base-learner 2. Similarly, it uses a 3D matrix $\mathbf{M}_i^{mT_1} = (M_i^m[1], M_i^m[2], \ldots, M_i^m[T_1])$ to record the mobile user distributions in the area of cell i's $n \times n$ adjacent cells during the first period. It applies a 3D sliding window with size $n \times n \times S_{b3}$ to split $\mathbf{M}_i^{mT_1}$ and generate the training set of $T_1 - S_{b3}$ base-samples for base-learner 3 by labelling each $n \times n \times S_{b3}$ matrix with cell i's WeChat traffic load in next time interval.

After the three base-learners are well-trained using the corresponding training sets, it generates one meta-sample related to each time interval in the second period by labelling the meta-features, which are represented by a vector consisting of the outputs of the base-learners and these outputs' credibilities, with cell i's ground-true WeChat traffic load in this time interval, and finally obtain the training set of T_2 meta-samples for the meta-learner in SLIM-TP. Finally, for mobile cell i, the prediction performance of SLIM-TP is tested using the WeChat traffic loads of this cell generated in the third period T_3.

5.4 Evaluations on Cellular Level WeChat Mobile Traffic Loads

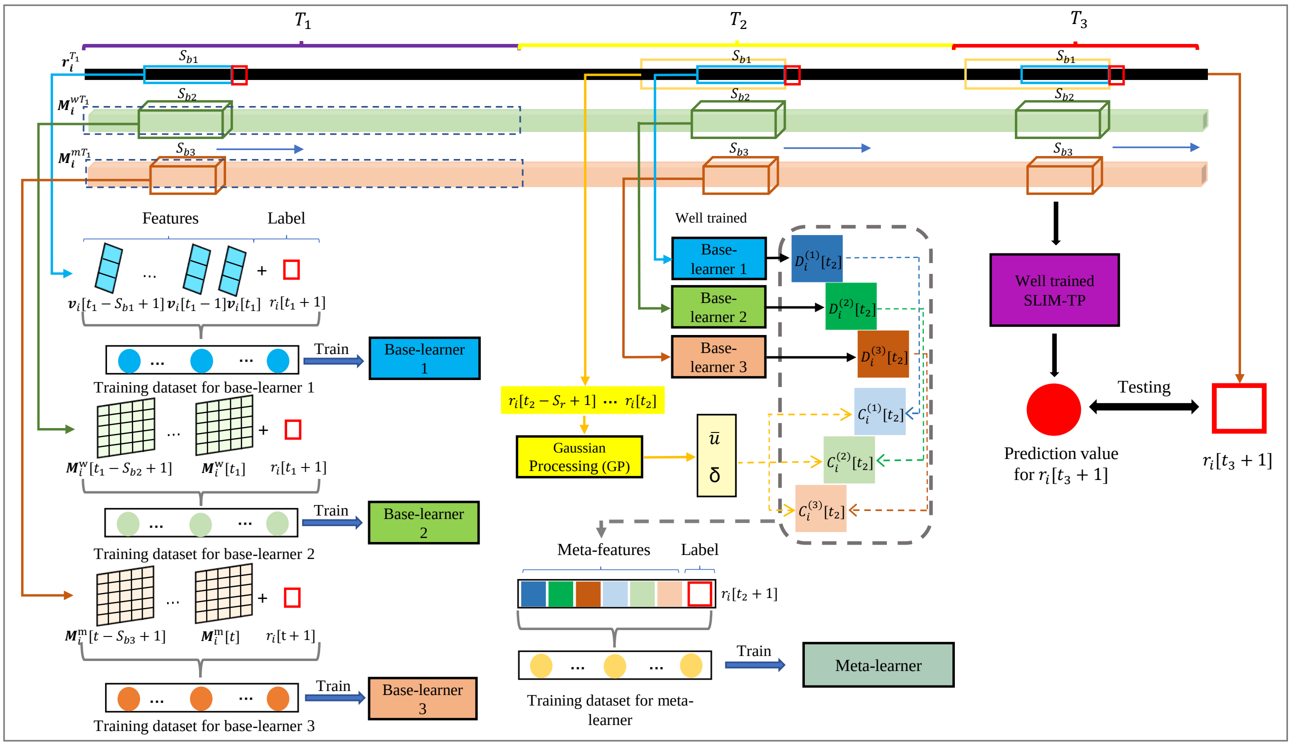


Fig. 5.8 The construction of training sets and testing sets for the base-learners and the meta-learner in SLIM-TP

5.4.1 Experimental Settings

This section evaluates the performance of the proposed framework using the dataset of real-world IM traffic records as described in Section II. In this experiments, L_{b1}, L_{b2}, and L_{b3} are set as 2, while S_{b1} and S_{b2} are set as 4 and S_{b3} is set as 6, respectively. For each target cell, the input matrices for base-learners 2 and 3 are with the same size of 21 × 21. The base-learner 1 has three layers, with 9, 16, and 24 neurons. And base-learner 2 has three layers with 16,32,16 filters, and the kernel is set a 3 × 3T. he first convolutional layer of base-learner 3 adopts 16 filters and the second layer adopts 32 filters, while the size of both kernels is set as 3 × 3 × 6. Specifically, batch normalisation and ReLu unit are used in base-learner 3 to accelerate the training process as well as prevent the overfitting problem. S_{rf} is set as 168. The kernel function in the GP-based referee is selected as a combination function which consists of radial-basis function (RBF) and the ExpSine squared kernel. The meta-learner has 3 hidden layers of sizes 32, 16, and 1, respectively. The batch size is 5 and the learning rate of 0.001. In practice, other structures of the meta-learner as well as the base-learners can be adopted in the proposed framework. However, finding the optimal structures for the learning models is outside the scope of this work.

The WeChat traffic loads and mobile user distributions related to a target mobile cell from 04/03/2019 to 17/03/2019 are used to construct the three base-learners' training sets and use the WeChat traffic loads of this cell from 18/03/2019 to 24/03/2019 to construct the meta-learner's training set. Finally, the traffic loads generated from 25/03/2019 to 31/03/2019 are utilized to test SLIM-TP's prediction accuracy.

The performance of SLIM-TP is compared with the existing mobile network traffic prediction methods including LR [135], ARIMA [129], SVR [136], and three deep learning-based prediction models that have the same structures with the three base-learners in SLIM-TP, respectively. Additionally, the chapter explores three excellent-performing traffic prediction models as baseline methods, each based on one of the three base-leaner of SLIM-TP.

- LSTM [59]. In this research, LSTM autoencoders are utilised for network data recognition and prediction. The model consists of four LSTM layers, with two layers serving as encoders and the other two layers functioning as decoders (predictors). It is one of the most classic LSTM models used for sequence prediction.

- ConvLSTM [60]. The research maximises the potential of ConvLSTM by designing an encoder and decoder that can simultaneously observe multiple time windows. By modelling traffic from different windows, the model can accelerate feature exploration in the temporal domain. Moreover, the convolutional part of ConvLSTM provides a good interpretation of spatial dependencies.

- 3D-CNN [61]. In this research, the unique characteristics of short-term prediction are leveraged to design a distinctive 3D-CNN application structure. By increasing the step size of the 3D-CNN, it can parallelly process each segment of data, reducing computational time while maintaining prediction performance.

These three traffic prediction models serve as baseline models for the proposed framework.

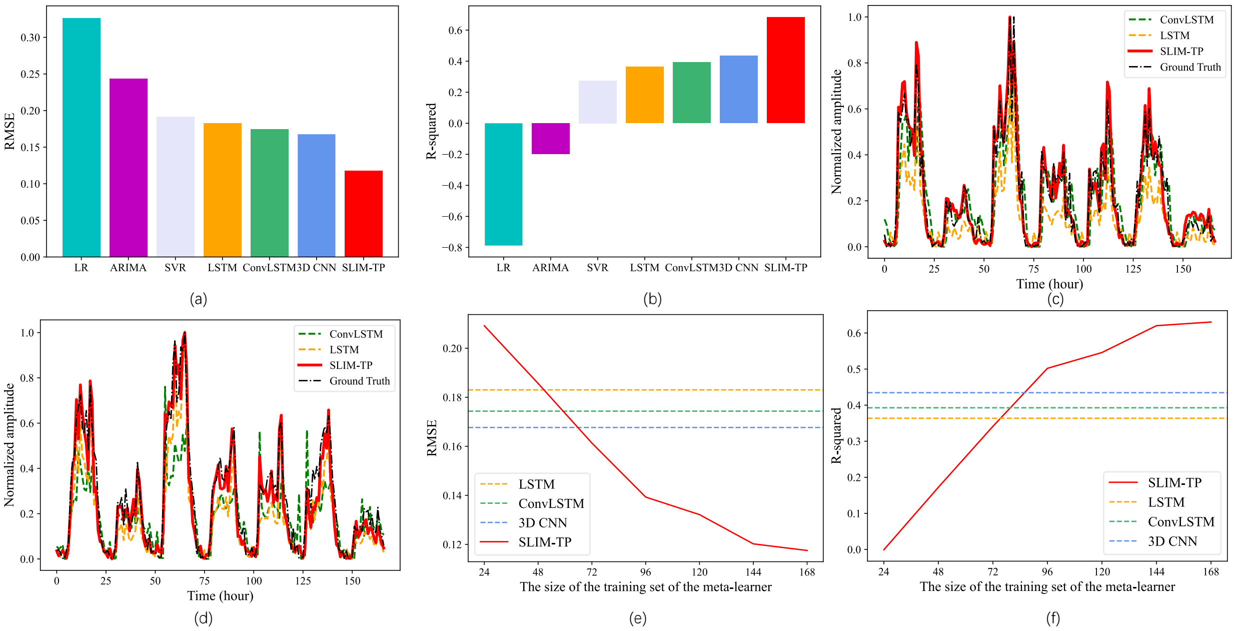
For a fair comparison, it trains these baseline methods for each target cell using the corresponding WeChat traffic loads and mobile user distributions generated from 04/03/2019 to 24/03/2019. The Mean Absolute Error (MAE) loss is used as the loss function, while the adaptive moment estimation (Adam) algorithm [173] with the default learning rate is utilized to optimise the baseline deep learning-based models as well as the meta-learner and base-learners in SLIM-TP. RMSE and R2 are adopted to evaluate the accuracy of the proposed framework as well as the baseline methods. R2 can be calculated as follows:

$$R2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{5.5}$$

, where $\bar{y}$ represents the mean value of the truth values. The coefficient of determination R2 takes values between -1 and 1, where a value closer to 1 indicates a better fit of the predictions to the truth values, and a value closer to -1 indicates a poorer fit. In other words, R2 measures how well the predicted values match the variability of the true values, with 1 being a perfect fit and 0 indicating no fit at all.

5.4.2 Prediction Performance

A comparison of the performance of SLIM-TP and the baseline methods, measured by the average RMSE and R2, is shown in Fig.5.9. From Fig.5.9(a) and Fig.5.9(b), it can be seen that ARIMA and LR perform the worst in predicting IM traffic among all the considered methods. This is because these two linear model-based methods are not capable of capturing the extremely nonlinear patterns in IM traffic loads. The SVR, which is a nonlinear and shadow learning-based prediction method, can handle the non-linearity in IM traffic variation and thus performs better than ARIMA and LR. Due to the ability of complex feature extraction, the three deep learning-based methods used in SLIM-TP are able to learn the deep dependencies between IM traffic loads generated in different time intervals, between IM traffic loads



(a) (b) (c) (d) (e) (f)

Fig. 5.9 The performance comparison between SLIM-TP and baselines: a) The RMSE results of proposed model and baselines on different grids; b) The R-squared value of proposed model and baselines; c) The ground true WeChat traffic loads generated in cell (57,32) from 25/03/2019 to 31/03/2019 as well as the predicted values of SLIM-TP and two baseline methods; d) The ground true WeChat traffic loads generated in cell (45,60) from 25/03/2019 to 31/03/2019 as well as the predicted values of SLIM-TP and two baseline methods; e) The SLIM-TP's RMSE performance vs. the size of meta-learner's training set; f) The SLIM-TP's R-squared performance vs. the size of meta-learner's training set

generated in adjacent mobile cells, or between the IM traffic variation and the historical mobile user distributions, and thus achieve lower RMSE values and higher R2 values than ARIMA, LR, and SVR. The proposed SLIM-TP obtains the best prediction accuracy among all the considered methods in terms of the lowest RMSE and the highest R2, attributed to two reasons. First, the three deep learning-based base-learners in SLIM-TP have the ability to learn and represent the complex spatiotemporal correlations of IM traffic loads as well as the deep dependency between the IM traffic variation and the mobile user distributions. Second, the meta-learner in SLIM-TP, which revises the outputs of the three base-learners according to their credibilities, makes SLIM-TP more adaptable to IM traffic patterns with obvious randomness and finally leads to highly reliable prediction results. Compared with the three baseline deep learning methods (LSTM, ConvLSTM, 3D CNN), SLIM-TP further reduces the RMSE by about 37 percent, 35 percent, and 31 percent, respectively, while improving the R2 by 88 percent, 74 percent, and 58 percent, respectively. Fig.5.9(c) and (d) show the prediction results of the SLIM-TP and the two conventional baseline deep learning based methods (LSTM and ConvLSTM) for two randomly selected mobile cells (57,32) and (45,60). Despite all three baseline models being multi-layer deep learning structures, each model has limitations in fully exploring the most effective features. LSTM is undoubtedly the weakest performer among them, mainly due to its focus on exploring only a single sequence. Although there have been attempts to flatten multidimensional information and input it into LSTM, the loss of spatial information and interference with temporal patterns may outweigh the benefits. ConvLSTM, with its capability to simultaneously explore spatiotemporal features, performs worse than the 3D-CNN network, which focuses on short-term prediction. This could be due to ConvLSTM's susceptibility to highly bursty traffic patterns. This proves the complexity of IM traffic, and that the schema of the data and the robustness of the system itself are extremely important. The 3D-CNN model outperforms the other baseline models, including LSTM and ConvLSTM, mainly because it analyses data using multiple segmented time periods. However, 3D-CNN has its own drawback of not being able to retain long-term information, which prevents it from becoming the core algorithm for traffic prediction.

It is worth noting that the three basic learning structures of SLIM-TP are simpler than the three baseline models, yet they achieve significantly better results. This highlights the importance of introducing highly correlated data as auxiliary information and the efficiency of the meta-learner in modelling multidimensional features.

All these results clearly show that the SLIM-TP achieves more accurate prediction values than the baseline methods, especially when the IM traffic pattern has abnormalities or sudden changes.

It also investigates how the stack learning technology, i.e., the meta-learner in SLIM-TP, can help the base-learners improve their prediction accuracy. Fig.5.9(e) and (f) show the average RMSE and R2 achieved by SLIM-TP over the mobile cells in Guangzhou versus the number of meta-samples used to train each meta-learner corresponding to a target cell in SLIM-TP. For comparison purpose, the average RMSE and R2 achieved by every base-learner are also given as constants. From Fig.5.9(e) and (f), it can be seen that SLIM-TP's prediction accuracy increases monotonously as the number of training meta-samples gets large and the performance of SLIM-TP will exceed that of each base-learner when the meta-sample number is greater than around 72. Such a finding can be intuitively explained as follows. When enough meta-samples are accumulated, the meta-learner will acquire the ability to extract the correlations between the constructed meta-features and the ground-true IM traffic load in next time interval for a given mobile cell. However, an interesting phenomenon existing in Fig.5.9(e) and (f) is that when the number of training meta-samples is larger than 144, the performance of SLIM-TP will increase little as the meta-learner's training set continues to augment. This finding implies that it does not need to construct excessively many meta-samples to train the meta-learner in SLIM-TP in practice.

5.4.3 Complexity Analyses

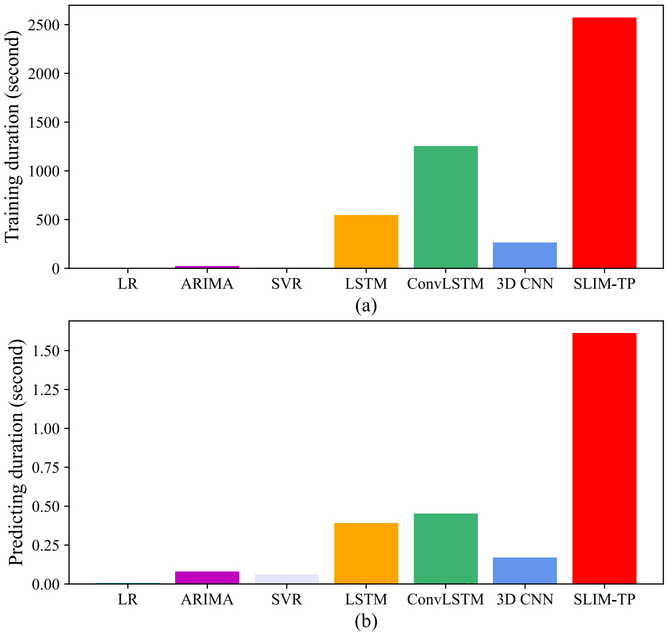


Fig. 5.10 The comparison of time consumption in training step and testing step between SLIM-TP and baselines: a) Off-line training time; b) On-line predicting time

The experiments are implemented by python (version 3.6.3) and programmed on the Windows 10 platform. The experimental computer is equipped with an i7 6700k CPU.

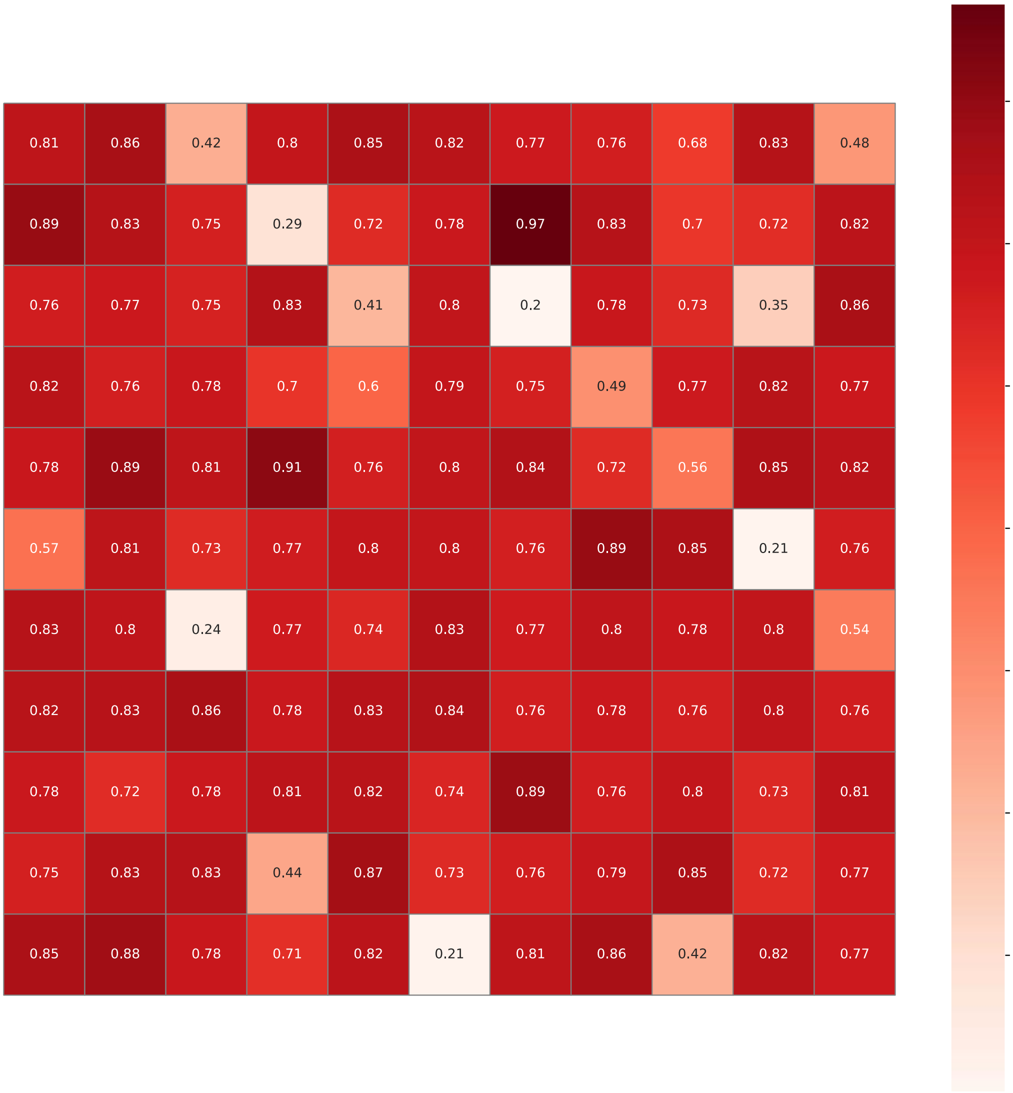
Tensorflow-GPU is also used to accelerate the training processes of neural networks through parallel computation.

In Fig.5.10, it demonstrates the average off-line training time needed by the proposed SLIM-TP and the baseline methods as well as the average on-line predicting time needed to output the prediction values in each time interval. From the graph, it can observe that the model-driven or shadow learning-based methods, i.e., the ARIMA, LR, and SVR, require relatively short off-line training time and on-line predicting time. On the contrary, the three baseline deep learning-based methods consume much longer time in both the off-line training and the on-line predicting than the ARIMA, LR, and SVR due to their complex structures. Consistent with intuition, the proposed SLIM-TP needs the longest off-line training time and on-line predicting time among all the considered methods. Note that since the three base-learners and the meta-learner in SLIM-TP only need to be trained once for each target mobile cell and the average on-line predicting time of SLIM-TP is still much shorter than the length of a time interval, SLIM-TP is applicable for practical IM traffic load prediction tasks.

5.5 The robustness of Proposed Model

To validate the robustness of SLIM-TP, it was applied to the training Milan dataset introduced in the previous chapter. To compensate for the missing grid population data, the high correlation between mobile network traffic and local calling frequencies was demonstrated. Fig.5.11 shows the correlation between traffic and phone calls within an 11×11 range. Therefore, calling frequencies can be used as auxiliary information to enhance the network traffic prediction algorithm.

In the previous chapter, MTL-STPN, MTL-STPN(no STPPC), and STCNet, with the ability to explore multidimensional data, are introduced as baseline models. The traffic data within a 4×4 region, encompassing sixteen grids, was used for the analysis. MTL-STPN with STPPC enables global search. RMSE was used as the prediction metric to compare the performance of each model.



0.81	0.86	0.42	0.8	0.85	0.82	0.77	0.76	0.68	0.83	0.48
0.89	0.83	0.75	0.29	0.72	0.78	0.97	0.83	0.7	0.72	0.82
0.76	0.77	0.75	0.83	0.41	0.8	0.2	0.78	0.73	0.35	0.86
0.82	0.76	0.78	0.7	0.6	0.79	0.75	0.49	0.77	0.82	0.77
0.78	0.89	0.81	0.91	0.76	0.8	0.84	0.72	0.56	0.85	0.82
0.57	0.81	0.73	0.77	0.8	0.8	0.76	0.89	0.85	0.21	0.76
0.83	0.8	0.24	0.77	0.74	0.83	0.77	0.8	0.78	0.8	0.54
0.82	0.83	0.86	0.78	0.83	0.84	0.76	0.78	0.76	0.8	0.76
0.78	0.72	0.78	0.81	0.82	0.74	0.89	0.76	0.8	0.73	0.81
0.75	0.83	0.83	0.44	0.87	0.73	0.76	0.79	0.85	0.72	0.77
0.85	0.88	0.78	0.71	0.82	0.21	0.81	0.86	0.42	0.82	0.77

Fig. 5.11 The correlation between traffic and calling data in Milan dataset

Fig.5.12 shows the performance of the four models. It can be observed that SLIM-TP performs slightly better than MTL-STPN, but the difference is not significant. MTL-STPN (no STPPC), as the only model without high-quality multidimensional auxiliary information, exhibits the poorest performance. It demonstrates that in mobile network traffic prediction, data with high correlation needs to be incorporated to enhance prediction performance. Comparing STCNet and SLIM-TP, the main difference lies not only in the choice of learning

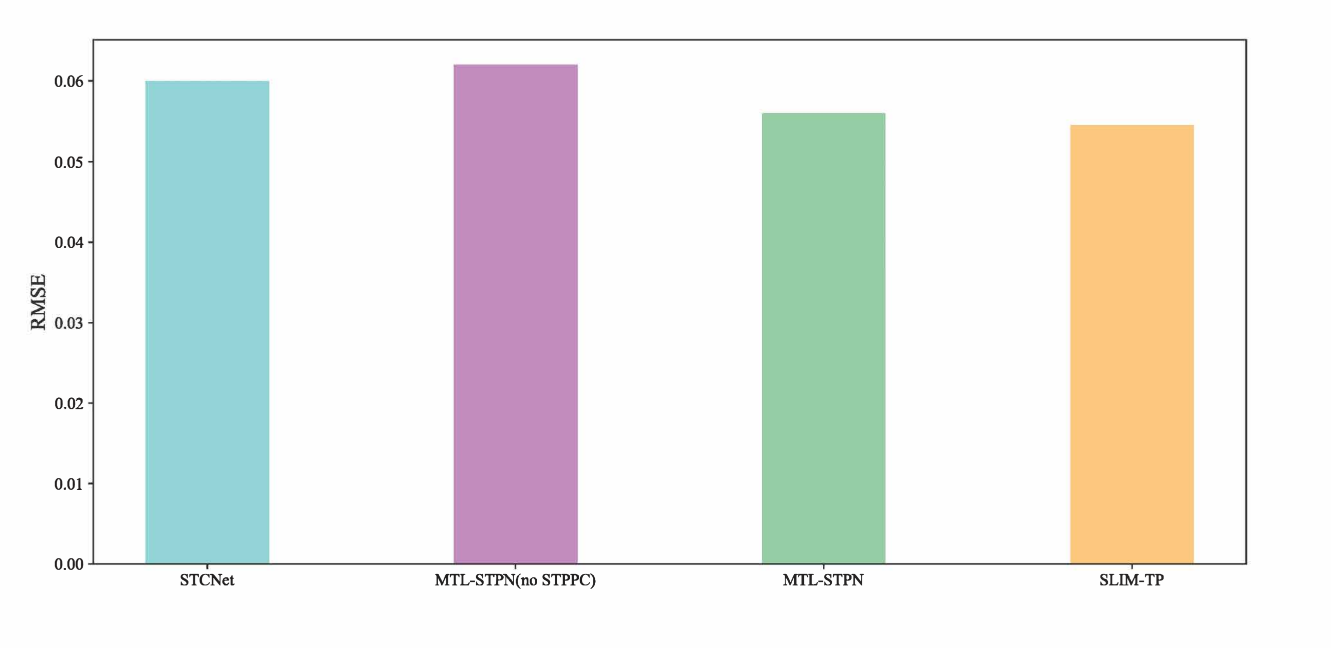


Fig. 5.12 The performance comparison between SLIM-TP and baselines training by Milan dataset

models but also in the application of the meta-learner in the framework. Processing features before modelling may result in information loss, whereas meta-learning directly models the features, enabling the extraction of global characteristics. Moreover, the quality of features trained by the meta-learner is crucial. It recalls to the need for filtering and selecting relevant information about the data and its auxiliary information. The limited difference between SLIM-TP and MTL-STPN may indicate that the feature modelling bottleneck has been reached. The information provided by similar traffic or related heterogeneous network data is also limited.

In conclusion, SLIM-TP demonstrates excellent performance in small-scale mobile network traffic prediction.

5.6 Conclusions

In this chapter, it makes an initiative attempt to introduce deep stack learning into mobile network traffic prediction and propose a novel deep stack learning based cellular-level

IM traffic prediction framework, SLIM-TP. Focusing on the IM traffic's autocorrelations, spatiotemporal correlations, and correlations with the mobile user distributions, three base-learners are designed in SLIM-TP to give the rough prediction values for the IM traffic load of a certain mobile cell. Taking the outputs of the base-learners and their corresponding credibilities as the meta-features, an MLP based meta-learner is used to further improve the reliability of prediction results of the IM traffic loads. Experiments with real-world IM traffic records demonstrate that SLIM-TP can elevate the prediction accuracy by about 31 percent to 37 percent as compared with the conventional deep learning based methods. Moreover, an investigation is also carried out on how the stack learning technology can improve the base-learners' performance so as to adjust or reduce the operational complexity of SLIM-TP.

The robustness of SLIM-TP is further validated for different types of mobile network traffic data. The proposed model is applicable to both cell-level and application-level data, and it outperforms other state-of-the-art models in terms of prediction accuracy. It verifies the effectiveness of the SLIM-TP framework in small-scale mobile network traffic prediction with auxiliary information applied.

Additionally, it is worth studying if some other characteristics of the IM traffic can be regarded as the meta-features in SLIM-TP.

Chapter 6

Conclusion & Future work

Overview

6.1 Conclusion

This thesis aims to predict mobile network traffic by focusing on mobile network user behaviour and traffic characteristics. The research is conducted using deep learning-based frameworks to enhance prediction performance. Mobile network traffic analysis and prediction have become one of the most important research fields in both academic and industry communities. This is not only due to the abundance of diverse data generated in complex network environments but also the challenges posed by the increasing complexity of network environments. Firstly, with the popularity of mobile networks among the population, the patterns of network traffic are becoming increasingly diverse. Many mathematical models that were once used to model traffic trends are no longer suitable for the current scenarios. This necessitates further research and exploration into the characteristics of the data. Secondly, due to advancements in network technology, the observation granularity of network traffic has increased, leading to greater randomness in traffic patterns. This requires more robust models to address data deficiencies. As a result, mobile network traffic prediction has become a highly discussed topic. In the face of increasingly complex traffic information and more demanding prediction requirements, researchers must determine which data features and technologies need to be applied.

The analysis of mobile network traffic can be divided into three directions: mobility analysis, network analysis, and social analysis. This thesis focuses on mobility analysis initially, exploring the periodicity of data and verifying the feasibility of using deep learning algorithms for sequence-related tasks. Subsequently, this thesis delves into network traffic prediction, where it conducts further research on the spatiotemporal characteristics of complex traffic patterns. As a result, a deep learning-based framework is designed, striking a balance between accuracy and efficiency. Finally, in the context of high granularity and strong randomness of network traffic, the thesis conducts comprehensive feature analysis and extraction. It proposes an algorithmic framework tailored to such features, aiming to achieve accurate predictions for complex traffic patterns. By undertaking these research efforts, it presents a method to effectively analyse and predict mobile network traffic.

Chapter 3 aims to predict mobility based on user historical data. The spatiotemporal characteristics of user mobility have always been a focus of mobility prediction, but there is no consensus on how to explore these features effectively. Through analysis, it was found that the frequent visitation locations of users often exhibit strong periodicity. Therefore, the chapter first identifies dense visitation areas for each user, and then analyses and measures the visitation frequency for each area rather than each user. Next, the PPM is applied to each reference region to obtain periodic information. Subsequently, based on the visitation records for each area at different time intervals, conditional probabilities are calculated to derive the POI preference for each user, consisting of their preference for each reference region. By incorporating visitation probabilities, the data dimensionality is increased. Furthermore, the chapter introduces LSTM, a neural network model suitable for sequence prediction, to predict the next destination for users. Through experimental comparisons, the proposed POI preference framework effectively enhances user feature modelling, resulting in a 30% improvement in prediction accuracy for the forecasting model. This approach also improves computational efficiency and reduces the complexity requirements of the model. Moreover, the study highlights the importance of feature analysis for sequential data, leading to a better understanding of mobility patterns and an enhancement in prediction accuracy.

Next, this thesis focuses on the prediction of mobile network traffic. Regarding densely base stations in the city, the correlation between different cell traffic has always been a topic of discussion. However, most research has only concentrated on learning the traffic around the target area. Inspired by the patterns of user periodic behaviour in the previous chapter, this chapter employs a correlation-based method to measure traffic correlations in distant regions and finds that distance is not the sole criterion for judging traffic correlation. Subsequently, considering the limitations of applying long-distance and large-scale spatiotemporal correlations, the MTL-STPN is proposed, which allows for training and prediction with shared feature parameters across different regions. Moreover, efficiently locating the target area becomes a new challenge. To address this, the chapter introduces a task detection method suitable for MTL-STPN, called STPPC, based on features such as correlation, similarity, and periodicity. This method improves the prediction efficiency and accuracy of MTL-STPN. Compared with state-of-the-art algorithms, the predictive performance can be enhanced by 10%.

Building upon the conclusions of Chapter 4, Chapter 5 delves further into the prediction of mobile network traffic with higher granularity and stronger bursts, focusing on application-level data, such as IM traffic. This chapter approaches the problem from a data-centric perspective and seeks to find highly correlated alternative data for traffic with strong bursts but weak periodicity. By employing conditional entropy and correlation coefficients, it gains an intuitive understanding of data correlations. Ultimately, it is revealed that the local population and IM traffic exhibit a strong correlation. Moreover, by applying the multi-task framework introduced in Chapter 4, it is discovered that training different data separately using parallel learners can fully exploit the advantages of the learners. Subsequently, the generated features are subjected to meta-learning, which models the obtained features. The trained meta-learner can then complement missing features, effectively improving the robustness of dealing with complex data. As a result, in experiments using real-world data for training, the predictive accuracy is improved by 30% compared to state-of-the-art models

6.2 Future work

In this thesis, the research conducted a lot of analysis on mobile network traffic and its related data, mainly on its temporal and spatial characteristics. Based on the analysis results and the characteristics of each type of data, it raises several possibilities to enrich the data by constructing auxiliary information that can be used to improve the performance of the system. The model of a personal POI based on periodic pattern exploration has been established. And for the prediction of high burstiness data, a stack deep learning framework is established through external high correlation information, spatiotemporal feature extraction, and unified nonlinear measurement. Finally, for the efficiency of spatial feature extraction, a multi-task learning framework has been proposed to exclude areas with low correlation information.

In Chapter 3, k-means has been used to cluster the location in the very first step. Since KDE is used in regional planning, the clustering method has not been discussed. In subsequent research, other clustering algorithms should be studied, as well as the classification algorithm under supervised learning. Secondly, the periodic complexity of the experimental data in the case study is not sufficient, and a more complex dataset needs to be found for verification. Finally, the pattern fusion algorithm needs more in-depth research to avoid being unable to cope with multiple periodic fusions.

In Chapter 4, STPPC is the part that can be improved. As proved in the final experiment, when the increase of the tasks exceeds the critical value, feature redundancy is likely to lead to system degradation. The more intuitive solution should be to use a threshold system so that once it finds that there are too many tasks or the task relevance is poor, the connection layer will be cut off, and the excluded parts can do STL on their own to ensure the average performance of the overall task. However, how to adapt the threshold of task relevance and how to make the redundancy do STL requires further research. It is difficult for the current methods to obtain such data through quantification. Transfer learning might be a potential solution such that through the migration of fine-tuned model parameters and the matching of the training data similarity, it might be possible to achieve threshold configuration or adjustment.

In Chapter 5, the referee module composed of Gaussian processes is worthy of further study, due to its being more like an unsupervised or semi-supervised learning method. In fact, other algorithms can also be applied, such as clustering algorithms to classify the results of the base learner. Secondly, meta-learner is also a part that can be further improved, such as joining deep learning, which needs to consider a trade-off strategy. As a deeper network normally implies a further increase in complexity, whether the gain on the performance improvement has obvious advantages over the loss risen from the increase in complexity requires further experiments to verify.

In addition to the above-mentioned approaches of system adjustment, the richness of the data is also one of the directions for future work. For example, in application-level data, whether the other application data have similar patterns is an open question. Finally, in data analysis, the similarity detection of spatiotemporal is also a point that needs continuous attention.

References

[1] C. Zhang, P. Patras and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," in *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224-2287, thirdquarter 2019

[2] D. Calin, H. Claussen and H. Uzunalioglu, "On femto deployment architectures and macrocell offloading benefits in joint macro-femto deployments," *IEEE Commun. Mag.*, vol. 48, no. 1, pp. 26-32, January 2010.

[3] B. Ma, W. Guo and J. Zhang, "A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning," *IEEE Access*, vol. 8, pp. 35606-35637, 2020.

[4] D. Martinez-Mosquera, R. Navarrete and S. Luján-Mora, "Development and Evaluation of a Big Data Framework for Performance Management in Mobile Networks," *IEEE Access*, vol. 8, pp. 226380-226396, 2020.

[5] Q. Sun, N. Li, J. Huang, X. Xu and Y. Xie, "Intelligent RAN Automation for 5G and Beyond," *IEEE Wirel. Commun.*, doi: 10.1109/MWC.014.2200271.

[6] Y. Wu, L. Ge, X. Yuan, X. Fu and M. Wang, "Adaptive Power Control Based on Double-layer Q-learning Algorithm for Multi-parallel Power Conversion Systems in Energy Storage Station," *Journal of Modern Power Systems and Clean Energy*, vol. 10.

[7] V. Hassija, V. Chamola, A. Agrawal, A. Goyal, N. Cong Luong, D. Niyato, F. Yu and M. Guizani, "Fast, Reliable, and Secure Drone Communication: A Comprehensive Survey," in *IEEE Commun. Surv. Tutor.*, vol. 23, no. 4, pp. 2802-2832, Fourthquarter 2021.

[8] Kamar, Hüveyscan, and Sema F. Oktuğ. "Communication Traffic Monitoring for Mobile Applications." *6th International Conference on Computer Science and Engineering (UBMK)*. 2021.

[9] B. Wen, Y. Qiao, W. Lin and J. Yang, "Connecting mobility, online behavior and urban structure from cellular network data," *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017.

[10] C. Williamson, E. Halepovic, H. Sun, and Y. Wu, “Characterization of CDMA2000 cellular data network traffic,” in *IEEE Conference on Local Computer Networks 30th Anniversary*, pp. Z000–719, IEEE, 2005.

[11] Z. Wang, J. Hu, G. Min, Z. Zhao, Z. Chang and Z. Wang, "Spatial-Temporal Cellular Traffic Prediction for 5G and Beyond: A Graph Neural Networks-Based Approach," in *IEEE Trans. Ind. Inform.*, vol. 19, no. 4, pp. 5722-5731, April 2023.

[12] S. He. S. Xiong, Y. Ou, J. Zhang, J. Wang and Y. Huang, "An Overview on the Application of Graph Neural Networks in Wireless Networks," in *IEEE open j. Commun. Soc.*, vol. 2, pp. 2547-2565, 2021.

[13] Chang, Wanjun, Dong Sun, and Qidong Du. "Intelligent Sensors for POI Recommendation Model Using Deep Learning in Location-Based Social Network Big Data" *Sensors*, 23, no. 2: 850, 2023.

[14] C. Wang, L. Ma, R. Li, T. S. Durrani and H. Zhang, "Exploring Trajectory Prediction Through Machine Learning Methods," in *IEEE Access*, vol. 7, pp. 101441-101452, 2019.

[15] T. Hu, S. Wang, B. She, M. Zhang, X. Huang and W. Guan “Human mobility data in the COVID-19 pandemic: characteristics, applications, and challenges”, *International Journal of Digital Earth*, 14:9, 1126-1147, 2021.

[16] Grantz, K.H., Meredith, H.R. and Cummings, D.A.T. “The use of mobile phone data to inform analysis of COVID-19 pandemic epidemiology”, *Nat Commun. 11*, 4961, 2020.

[17] A. Wang, A. Zhang, Edwin H. W. Chan, Wenzhong Shi, Xiaolin Zhou, and Zhewei Liu. "A Review of Human Mobility Research Based on Big Data and Its Implication for Smart City Development" *ISPRS International Journal of Geo-Information 10*, no. 1: 13, 2021.

[18] Y. Qiao, Z. Xing, Z. M. Fadlullah, J. Yang and N. Kato, "Characterizing Flow, Application, and User Behavior in Mobile Networks: A Framework for Mobile Big Data," in *IEEE Wirel. Commun.*, vol. 25, no. 1, pp. 40-49, February 2018.

[19] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico and A. Pescapé, "Characterization and Prediction of Mobile-App Traffic Using Markov Modeling," in *IEEE TNSM*, vol. 18, no. 1, pp. 907-925, March 2021.

[20] Y. Zhou, R. Li, Z. Zhao, X. Zhou and H. Zhang, "On the α-Stable Distribution of Base Stations in Cellular Networks," in *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1750-1753, Oct. 2015.

[21] S. Escriche, E. Vassaki, and Peters, “A comparative study of cellular traffic prediction mechanisms”. *Wireless Netw 29*, 2371–2389, 2023.

[22] W. Jiang, "Cellular traffic prediction with machine learning: A survey." *Expert Systems with Applications* 201: 117163, 2022.

[23] X., Jun, C. Yang, and Y. Yuan. Channel Coding in 5G New Radio. *CRC Press*, 2022.

[24] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini and J. Widmer, "A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques," in *IEEE Commun. Surv.*, vol. 19, no. 3, pp. 1790-1821, thirdquarter 2017.

[25] Alfaia RD, Souto AVdF, Cardoso EHS, Araújo JPLd and Francês CRL. “Resource Management in 5G Networks Assisted by UAV Base Stations: Machine Learning for Overloaded Macrocell Prediction Based on Users’ Temporal and Spatial Flow”. *Drones.* , 6(6):145, 2022.

[26] K. Suh, S. Kim, Y. Ahn, S. Kim, H. Ju and B. Shim, "Deep Reinforcement Learning-Based Network Slicing for Beyond 5G," in *IEEE Access*, vol. 10, pp. 7384-7395, 2022.

[27] Alrowais, Fadwa and R. Marzouk, "Artificial Intelligence Based Data Offloading Technique for Secure MEC Systems." *Computers, Materials Continua 72.2* ,2022.

[28] Shambhavi, A. Prakash, M. K and A. K. Joseph, "Prediction of Load Demand on Base Stations for application of Energy Conservation Using Deliberated Informer," *2023 International Conference on Microwave, Optical, and Communication Engineering (ICMOCE)*, Bhubaneswar, India, 2023.

[29] K. Xu, R. Singh, H. Bilen, M. Fiore, M. K. Marina and Y. Wang, "CartaGenie: Context-Driven Synthesis of City-Scale Mobile Network Traffic Snapshots," *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Pisa, Italy, 2022.

[30] L. Zhu, B. Zhaoand Y. An, “A stacked broad learning system with multitask learning method for cellular wireless network traffic prediction”. *Soft Comput* , 2022.

[31] Z. Wang and V. W. S. Wong, "Cellular Traffic Prediction Using Deep Convolutional Neural Network with Attention Mechanism," *IEEE ICC 2022*, Seoul, Korea, Republic of, 2022.

[32] K. Zhang, G. Chuai, J. Zhang, X. Chen, Z. Si and S. Maimaiti, “DIC-ST: A Hybrid Prediction Framework Based on Causal Structure Learning for Cellular Traffic and Its Application in Urban Computing”. *Remote Sensing*, 2022.

[33] X. Zhou, Y. Zhang, and Z. Li, “Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning”, *Neural Comput Applic 34*, 5549–5559 2022.

[34] G. Alam, "Curating Datasets from GPS, Communication Technology and Social Media: Using Artificial Intelligence to Predict, Analyse and Manage Traffic System," *2022 2nd ICCIT*, Tabuk, Saudi Arabia, 2022

[35] T. LiT. Xia, H Wang, Z. Tu, S. Tarkoma, Z. Han and P. Hui, "Smartphone App Usage Analysis: Datasets, Methods, and Applications," in *IEEE Commun. Surv.*, vol. 24, no. 2, pp. 937-966, Secondquarter 2022

[36] I. Trestian, S. Ranjan, A. Kuzmanovic and A. Nucci, "Measuring serendipity: connecting people, locations and interests in a mobile 3G network." *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, 2009.

[37] F. Sun, P. Wang, J. Zhao, N. Xu and J. Zeng, "Mobile Data Traffic Prediction by Exploiting Time-Evolving User Mobility Patterns," in *IEEE Trans. Mob. Comput.*, vol. 21, no. 12, pp. 4456-4470, 1 Dec. 2022.

[38] X. Pan, X. Cai, K. Song, T. Baker, T. R. Gadekallu and X. Yuan, "Location Recommendation Based on Mobility Graph With Individual and Group Influences," in *IEEE Trans. Intell. Transp. Syst.*, 2022.

[39] B. Ma, B. Yang, Y. Zhu and J. Zhang, "Context-Aware Proactive 5G Load Balancing and Optimization for Urban Areas," in *IEEE Access*, vol. 8, pp. 8405-8417, 2020.

[40] N. Tempelmeier, Y. Rietz, I. Lishchuk and T Kruegel, "Data4urbanmobility: Towards holistic data analytics for mobility applications in urban regions." *Companion Proceedings of The 2019 World Wide Web Conference*, 2019.

[41] J. Suzuki, Y. Suhara, H. Toda and K Nishida, "Personalized visited-poi assignment to individual raw GPS trajectories." *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 5.3 , 2019.

[42] Wang, Di, T. Miwa, and T. Morikawa. "Big trajectory data mining: a survey of methods, applications, and services." *Sensors*, 20.16 : 4571., 2020.

[43] E. Toch, B. Lerner, E. Ben-Zion and I Ben-Gal "Analyzing large-scale human mobility data: a survey of machine learning methods and applications." *Knowledge and Information Systems*, 58: 501-523., 2019.

[44] H. Tong, T. Wang, Y. Zhu, X. Liu, S. Wang and C. Yin, "Mobility-Aware Seamless Handover With MPTCP in Software-Defined HetNets," in *IEEE TNSM*, vol. 18, no. 1, pp. 498-510, March 2021.

[45] Fuchs, Matthias, and W. Höpken. "Clustering: Hierarchical, k-Means, DBSCAN." Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications." *Cham: Springer International Publishing*, 129-149. 2022.

[46] H. Rizk, A. Elmogy and H. Yamaguchi." A Robust and Accurate Indoor Localization Using Learning-Based Fusion of Wi-Fi RTT and RSSI". *Sensors*. 2022.

[47] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, vol. 13, pp. 2605–2611, 2013.

[48] C. Li, D. Li, Z. Zhang and D. Chu. "MST-RNN: A Multi-Dimension Spatiotemporal Recurrent Neural Networks for Recommending the Next Point of Interest". *Mathematics*. 2022

[49] Kong, D. jiang, and F. Wu. "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction." *IJCAI.* Vol. 18. No. 7. 2018.

[50] Y. Tian, P. Liu, J. Su and X. Ding, "Crowdsensing based missing data inference algorithm considering outlier data and GPS errors." *Information Sciences*, 612, 2022.

[51] Awawdeh, Shatha, H. Faris, and H. Hiary, "EvoImputer: An evolutionary approach for Missing Data Imputation and feature selection in the context of supervised learning." *Knowledge-Based Systems*, 236: 107734, 2022

[52] F. A. Vinisha and L. Sujihelen, "Study on Missing Values and Outlier Detection in Concurrence with Data Quality Enhancement for Efficient Data Processing," *2022 4th ICSSIT*, Tirunelveli, India, 2022.

[53] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining correlation between locations using human location history,” in *The 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 472–475, 2009.

[54] Angkhawey, Uriwan, and V. Muangsin, "Detecting points of interest in a city from taxi gps with adaptive dbscan." *2018 ICT-ISPC*. IEEE, 2018.

[55] Rendle, Steffen, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation." *The 19th international conference on World wide web*, 2010.

[56] P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, VS. Sheng and X. Zhou, "Where to go next: A spatio-temporal gated network for next poi recommendation." *IEEE Trans. Knowl. Data Eng.*, 34.5: 2512-2524, 2020.

[57] A. Montieri, G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, "Packet-level prediction of mobile-app traffic using multitask deep learning." *Computer Networks*, 200: 108529, 2021.

[58] W. R. Tobler, “A computer movie simulating urban growth in the detroit region,” *Econ. Geogr.*, vol. 46, no. 1, pp. 234–240, Jun. 1970.

[59] Nguyen, T. Anh, and P. Martins. "Cellular traffic type recognition and prediction." *2021 IEEE 32nd PIMRC*. IEEE, 2021.

[60] Y. Li, S. Chai, G. Wang, X. Zhang and J. Qiu, "Quantifying the uncertainty in long-term traffic prediction based on PI-ConvLSTM network." *IEEE Trans. Intell. Transp. Syst.*, 23.11: 20429-20441, 2022.

[61] F. Yu, D. Wei, S. Zhan and Y Shao, "3D CNN-based accurate prediction for large-scale traffic flow." *2019 4th ICITE*. IEEE, 2019.

[62] Cisco and S. Jose, “Cisco visual networking index (VNI) global mobile data traffic forecast update, 2017-2022 white paper,” Ca, Usa, pp. 3–5, 2019, [Online]. Available: http://www.gsma.com/spectrum/wp-content/uploads/2013/03/Cisco_VNI-global-mobile-data-traffic-forecast-update.pdf.

[63] Cisco, “Cisco Annual Internet Report (2018–2023),” *Cisco*, pp. 1–41, 2020, [Online]. Available: http://grs.cisco.com/grsx/cust/grsCustomerSurvey.html?SurveyCode=4153ad_id=US-BN-SEC-M-CISCOASECURITYRPT-ENTKeyCode=000112137.

[64] B. Yang, W. Guo, B. Chen, G. Yang, and J. Zhang, “Estimating Mobile Traffic Demand Using Twitter,” *IEEE Wireless Commun.*, vol. 5, no. 4, pp. 380–383, 2016.

[65] D. Naboulsi, M. Fiore, S. Ribot and R. Stanica, "Large-Scale Mobile Traffic Analysis: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 124-161, First-quarter 2016.

[66] B. Ma, W. Guo, and J. Zhang, “A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning,” *IEEE Access*, vol. 8, pp. 35606–35637, 2020.

[67] W. Sun, D. Miao, X. Qin and G. Wei, "Characterizing User Mobility from the View of 4G Cellular Network," *IEEE International Conference on MDM*, pp. 34-39, 2016.

[68] Y. Ouyang, Z. Li, L. Su, W. Lu, and Z. Lin, “Application Behaviors Driven Self-Organizing Network (SON) for 4G LTE Networks,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 3–14, 2020.

[69] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, “A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2392–2431, 2017.

[70] H. Mehdi, Z. Pooranian, and P. G. Vinueza Naranjo, “Cloud traffic prediction based on fuzzy ARIMA model with low dependence on historical data,” *Trans. ETT.*, no. May, pp. 1–17, 2019.

[71] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, “Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data,” *IEEE JSAC*, vol. 37, no. 6, pp. 1389–1401, 2019.

[72] C. Marquez, A. Banchs, M. Gramaglia, C. Ziemlicki, M. Fiore, and Z. Smoreda, “Not all apps are created equal: Analysis of spatiotemporal heterogeneity in nationwide mobile service usage,” *CoNEXT ’17*, pp. 180–186, 2017.

[73] “Digital 2020: Global Digital Overview — DataReportal – Global Digital Insights.”, May, 2021 [Online]. https://datareportal.com/reports/digital-2020-global-digital-overview.

[74] C. M. Krause and L. Zhang, “Short-term travel behavior prediction with GPS, land use, and point of interest data,” *TRANSPORT RES B-METH*, vol. 123, pp. 349–361, 2019.

[75] S. Sun, J. Chen, and J. Sun, “Traffic congestion prediction based on GPS trajectory data,” *Int. J. Distrib. Sens. Netw.*, vol. 15, no. 5, 2019.

[76] R. W. Douglass, D. A. Meyer, M. Ram, D. Rideout, and D. Song, “High resolution population estimates from telecommunications data,” *EPJ Data Sci.*, vol. 4, no. 1, pp. 1–13, 2014.

[77] H. Salat, Z. Smoreda, and M. Schlapfer, “A method to estimate population densities and electricity consumption from mobile phone data in developing countries,” *PLoS One*, vol. 15, no. 6 june, pp. 1–11, 2020.

[78] G. Khodabandelou, V. Gauthier, M. El-Yacoubi, and M. Fiore, “Population estimation from mobile network traffic metadata,” *WoWMoM 2016*, pp. 1–9, 2016.

[79] H. D. Trinh, L. Giupponi, and P. Dini, “Mobile Traffic Prediction from Raw Data Using LSTM Networks,” *IEEE PIMRC 2018*, vol. 2018-Septe, pp. 1827–1832, 2018.

[80] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[81] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, “Spatio-Temporal Wireless Traffic Prediction with Recurrent Neural Network,” *IEEE Wirel. Commun. Lett.*, vol. 7, no. 4, pp. 554–557, 2018.

[82] H. D. Trinh, N. Bui, J. Widmer, L. Giupponi, and P. Dini, “Analysis and modeling of mobile traffic using real traces,” *IEEE PIMRC 2017*, pp. 1–6, October, 2017.

[83] Zhou K, Tian Z, Yang Y. “Periodic pattern detection algorithms for personal trajectory data based on spatiotemporal multi-granularity,“ *IEEE Access*, vol. 7, pp. 99683-99693, 2019.

[84] K. Soltani Naveh and J. Kim, "Urban Trajectory Analytics: Day-of-Week Movement Pattern Mining Using Tensor Factorization," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2540-2549, July 2019.

[85] Georgiou H, Karagiorgou S, Kontoulis Y, et al. "Moving objects analytics: survey on future location trajectory prediction methods," *arXiv preprint*, pp. 1-43, 2018.

[86] J. Han, Z. Li, L. Tang, “Mining Moving Object and Traffic Data,”. *DASFAA 2010 Tutorial*, pp: 29, 2010.

[87] Y. Ye, Y. Zheng, Y. Chen, J. Feng and X. Xie, "Mining Individual Life Pattern Based on Location History," *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pp. 1-10, 2009. A. Wahid, M. T. U. Haider, M. M. Ahmad and A. K. Singh, "Adaptive route planning in spatial road networks," emph2016 WiSPNET, pp. 1281-1285, 2016.

[88] M. Zeng *et al.*, "Temporal-Spatial Mobile Application Usage Understanding and Popularity Prediction for Edge Caching," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 36-42, JUNE 2018.

[89] Q. Zeng, Q. Sun, G. Chen, H. Duan, C. Li and G. Song, "Traffic Prediction of Wireless Cellular Networks Based on Deep Transfer Learning and Cross-Domain Data," *IEEE Access*, vol. 8, pp. 172387-172397, 2020.

[90] X. Wang *et al.*, "Spatio-Temporal Analysis and Prediction of Cellular Traffic in Metropolis," *IEEE Trans. Mob. Comput.*, vol. 18, no. 9, pp. 2190-2202, 1 Sept. 2019.

[91] Cao H., Cheung D.W. and Mamoulis N. "Discovering Partial Periodic Patterns in Discrete Data Sequences," *Pacific-Asia conference on knowledge discovery and data mining*, vol 3056, 2004.

[92] Vlachos, M., Yu, P. and Castelli, V. "On periodicity detection and structural periodic similarity," *2005 Proc. SIAM Int. Conf. Data Min.*, pp. 449-460, April, 2005.

[93] R. Uday Kiran, C. Saideep, K. Zettsu, M. Toyoda, M. Kitsuregawa and P. Krishna Reddy, "Discovering Partial Periodic Spatial Patterns in Spatiotemporal Databases," *2019 IEEE Int. Conf. Big Data*, pp. 233-238, 2019.

[94] A. Telikani, A. H. Gandomi and A. Shahbahrami, "A survey of evolutionary computation for association rule mining," *Inf. Sci. (Ny).*, vol. 524, pp. 318–352, Jul. 2020.

[95] D. Zhang, K. Lee and I. Lee, "Periodic pattern mining for spatio-temporal trajectories: A survey," *ISKE 2015*, pp. 306–313, 2016.

[96] F. Giannotti, M. Nanni, F. Pinelli and D. Pedreschi, "Trajectory pattern mining," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 330–339, 2007. [Online]. doi: 10.1145/1281192.1281230.

[97] G. Xu, S. Gao, M. Daneshmand, C. Wang and Y. Liu, "A Survey for Mobility Big Data Analytics for Geolocation Prediction," *IEEE Wirel. Commun.*, vol. 24, no. 1, pp. 111–119, 2017. [Online]. doi: 10.1109/MWC.2016.1500131WC.

[98] S. A. Ahmed, D. P. Dogra, S. Kar and P. P. Roy, "Trajectory-Based Surveillance Analysis: A Survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1985–1997, 2019. [Online]. doi: 10.1109/TCSVT.2018.2857489.

[99] G. N. V. G. Sirisha, M. Shashi and G. V. P. Raju, "Periodic Pattern Mining – Algorithms and Applications," *Glob. J. Comput. Sci. Technol. Softw. Data Eng.*, vol. 13, no. 13, 2013.

[100] D. Zhang, K. Lee, and I. Lee, “Semantic periodic pattern mining from spatio-temporal trajectories,” *Inf. Sci. (Ny).*, vol. 502, pp. 164–189, 2019. [Online]. doi: 10.1016/j.ins.2019.06.035.

[101] [1] M. Zhou, Y. Wang, Y. Liu and Z. Tian, “An Information-Theoretic View of WLAN Localization Error Bound in GPS-Denied Environment,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4089–4093, 2019. [Online]. doi: 10.1109/TVT.2019.2896482.

[102] S. Mao, E. Tu, G. Zhang, L. Rachmawati, E. Rajabally and G.-B. Huang, “An Automatic Identification System (AIS) Database for Maritime Trajectory Prediction and Data Mining,” *Proceedings of ELM-2016*, pp. 241–257, 2018. [Online]. doi: 10.1007/978-3-319-57421-$9_2$0.*X.Chen*,*D.Shi*,*B.Zhao*,*andF.Liu*,ı*PeriodicPatternMiningBasedonGPSTrajectories*,▮*ISAEECE 20*
10.2991/*isaeece* – 16.2016.36.

[104] B. J. Worton, “Kernel Methods for Estimating the Utilization Distribution in Home-Range Studies,” *Ecology*, vol. 70, no. 1, pp. 164–168, Feb. 1989. [Online]. doi: 10.2307/1938423.

[105] Z. Li and J. Han, “Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data,” *Data Mining and Knowledge Discovery for Big Data*, pp. 41–81, 2014. [Online]. doi: 10.1007/978-3-642-40837-3_2.*K.J.Yang*,*T.P.Hong*,*Y.M.ChenandG.C.Lan*,ı*Projection* – *basedpartialperiodicpatternminingforeventsequences*,▮*Expert Syst. Appl.*,*vol*.40,*no*.10,*pp*.4232˘4240,*Aug*
10.1016/*J.ESWA*.2013.01.021.

[107] P. Huang, C.J. Liu, X. Yang, L. Xiao and J. Chen, "Wireless Spectrum Occupancy Prediction Based on Partial Periodic Pattern Mining," *IEEE TPDS*, vol. 25, no. 7, pp. 1925-1934, July 2014. [Online]. doi: 10.1109/TPDS.2013.283.

[108] D. Zhang, K. Lee, and I. Lee, “Hierarchical trajectory clustering for spatio-temporal periodic pattern mining,” *Expert Syst. Appl.*, vol. 92, pp. 1–11, 2018. [Online]. doi: 10.1016/j.eswa.2017.09.040.

[109] Q. Yuan, J. Shang, X. Cao, C. Zhang, X. Geng and J. Han, “Detecting multiple periods and periodic patterns in event time sequences,” *Int. Conf. Inf. Knowl. Manag. Proc.*, vol. Part F1318, pp. 617–626, 2017. [Online]. doi: 10.1145/3132847.3133027.

[110] J. Dezert, A. Tchamova, and D. Han, “Total Belief Theorem and Generalized Bayes’ Theorem,” *FUSION 2018*, no. 1, pp. 1040–1047, 2018. [Online]. doi: 10.23919/ICIF.2018.8455351.

[111] S. Akoush and A. Sameh, “The use of Bayesian learning of neural networks for mobile user position prediction,” *ISDA 2007*, pp. 441–446, 2007. [Online]. doi: 10.1109/ISDA.2007.4389648.

[112] M. A. Galal, W. M. Hussein, E. El-Din Abdelkawy and M. M. A. Sayed, “Satellite battery fault detection using Naïve Bayesian classifier,” *IEEE Aerosp. Conf. Proc.*, pp. 1–11, 2019. [Online]. doi: 10.1109/AERO.2019.8741963.

[113] D. Clemente, G. Soares, D. Fernandes, R. Cortesao, P. Sebastiao and L. S. Ferreira, “Traffic forecast in mobile networks: Classification system using machine learning,” *IEEE Veh. Technol. Conf.*, 2019. [Online]. doi: 10.1109/VTCFall.2019.8891348.

[114] S. Mehrizi, A. Tsakmalis, S. Chatzinotas and B. Ottersten, “A Feature-Based Bayesian Method for Content Popularity Prediction in Edge-Caching Networks,” *IEEE WCNC*, Apr. 2019. [Online]. doi: 10.1109/WCNC.2019.8885590.

[115] T. Tanaka, S. Ata and M. Murata, “Analysis of popularity pattern of User Generated contents and its application to content-aware networking,”*IEEE GC Wkshps 2016*, 2016. [Online]. doi: 10.1109/GLOCOMW.2016.7848837.

[116] E. S. Han and A. goleman, “International Journal of Advanced Research in Artificial Intelligence,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[117] B. Ma, W. Guo, and J. Zhang, “A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning,” *IEEE Access*, vol. 8, pp. 35606–35637, 2020. [Online]. doi: 10.1109/ACCESS.2020.2975004.

[118] Y. Singh and A. Mohan, “A Survey on Unsupervised Clustering Algorithm based on K-Means Clustering,” *Int. J. Comput. Appl.*, vol. 156, no. 8, pp. 6–9, 2016. [Online]. doi: 10.5120/ijca2016912481.

[119] W. Sun, X. Qin and G. Wei, “Modeling and mining the temporal patterns of service in cellular network,” *China Commun.*, vol. 12, no. 9, pp. 11–21, 2015. [Online]. doi: 10.1109/CC.2015.7275255.

[120] Z. M. Nia and M. R. Khayyambashi, “Improving content popularity prediction with k-means clustering and deep-belief networks,” *Multimed. Tools Appl.* vol. 80, no. 10, pp. 15745–15764, Feb. 2021. [Online]. doi: 10.1007/S11042-020-10463-X.

[121] C. Zhang, S. Dang, B. Shihada and M. S. Alouini, “Dual attention-based federated learning for wireless traffic prediction,” *IEEE INFOCOM*, 2021. [Online]. doi: 10.1109/INFOCOM42981.2021.9488883.

[122] Y. Wang and J. Ren, “Taxi Passenger Hot Spot Mining Based on a Refined K-Means++ Algorithm,” *IEEE Access*, vol. 9, pp. 66587–66598, 2021. [Online]. doi: 10.1109/ACCESS.2021.3075682.

[123] D. Srinivas and E. V Prasad, “Adaptive Density-Based Localization Algorithm Using Particle Swarm Optimization and DBSCAN Clustering Approach,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 11, pp. 5053–5062, 2021.

[124] M. Sheikh, S. M. Mashuda, R. Abedin and M. O. Rahman, “Dynamic Topology Reconstruction on Next Generation WLAN Using Spatial Reuse Gain by DBSCAN Clustering Algorithm,” *BIOM 2021*, pp. 315–327, 2022. [Online]. doi: 10.1007/978-981-16-6636-$0_25.D.M.Woo,D.C.Park,Y.S.Song,Q.D.NguyenandQ.D.N.Tran,{}_{1}Terrainclassificationusingclusteringalgor$
10.1109$/ICNC$.2007.705.

[125][126] Chakraborty S, Nagwani N K and Dey L. "Performance comparison of incremental k-means and incremental dbscan algorithms," *arXiv preprint*,Volume 27,2011.[Online]. arXiv:1406.4751, 2014.

[127] H. Aftab, J. Shuja, W. Alasmary and E. Alanazi, "Hybrid DBSCAN based Community Detection for Edge Caching in Social Media Applications," *2021 IWCMC*, pp. 2038-2043, 2021. [Online]. doi: 10.1109/IWCMC51323.2021.9498609.

[128] Solomonoff, Raymond J. "An inductive inference machine." *IRE Convention Record*, Vol. 2, pp. 56-62, 1957.

[129] Q. Yu, L. Jibin and L. Jiang, "An Improved ARIMA-Based Traffic Anomaly Detection Algorithm for Wireless Sensor Networks," *Int. J. Distrib. Sens. Networks*, 2016. [Online]. doi: 10.1155/2016/9653230.

[130] Y. Yu, J. Wang, M. Song and J. Song, "Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient," *ISDEA 2010*, vol. 1, no. 1, pp. 980–983, 2010. [Online]. doi: 10.1109/ISDEA.2010.335.

[131] H. Sheng, Q. Yan and K. Li, "Alpha Stable Distribution Based FARIMA Modeling and Forecasting for Network Traffic Data," *J. Phys. Conf. Ser.*, Vol. 1574, No. 1, p. 012135, 2020. [Online]. doi: 10.1088/1742-6596/1574/1/012135.

[132] F. Ju, J. Yang and H. Liu, "Analysis of Self-Similar Traffic Based on the On/Off Model," *2009 IWCFTA*, pp. 301-304, 2009. [Online]. doi: 10.1109/IWCFTA.2009.69.

[133] R. Li, Z. Zhao, X. Zhou, J. Palicot and H. Zhang, "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, 2014. [Online]. doi: 10.1109/MCOM.2014.6829969.

[134] M. E. Morocho-Cayamcela, H. Lee and W. Lim, "Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions," *IEEE Access*, vol. 7, pp. 137184–137206, 2019. [Online]. doi: 10.1109/ACCESS.2019.2942390.

[135] H. Sun, H. X. Liu, H. Xiao, R. R. He and B. Ran, "Use of Local Linear Regression Model for Short-term Traffic Forecasting," *Transp. Res. Rec.*, no. 1836, pp. 143–150, 2003. [Online]. doi: 10.3141/1836-18.

[136] Y. Liu and R. X. Wang, “Study on network traffic forecast model of SVR optimized by GAFSA,” *Chaos, Solitons and Fractals*, vol. 89, pp. 153–159, 2016. [Online]. doi: 10.1016/j.chaos.2015.10.019.

[137] N. Ramakrishnan and T. Soni, “Network Traffic Prediction Using Recurrent Neural Networks,” *ICMLA 2018*, pp. 187–193, 2019. [Online]. doi: 10.1109/ICMLA.2018.00035.

[138] C. Zhang, H. Zhang, D. Yuan and M. Zhang, “Citywide Cellular Traffic Prediction Based on Densely Connected Convolutional Neural Networks,” *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656–1659, 2018. [Online]. doi: 10.1109/LCOMM.2018.2841832.

[139] C. Zhang, M. Fiore and P. Patras, “Multi-Service Mobile Traffic Forecasting via Convolutional Long Short-Term Memories,” *MN 2019*, pp. 1-6, 2019.

[140] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016. [Online]. doi: 10.1109/CVPR.2016.90.

[141] R. Madan and P. Sarathimangipudi, “Predicting Computer Network Traffic: A Time Series Forecasting Approach Using DWT, ARIMA and RNN,” *IC3 2018*, no. 1, pp. 2–4, 2018. [Online]. doi: 10.1109/IC3.2018.8530608.

[142] K. Zhang, Z. Liu, and L. Zheng, “Short-Term Prediction of Passenger Demand in Multi-Zone Level: Temporal Convolutional Neural Network with Multi-Task Learning,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1480–1490, 2020. [Online]. doi: 10.1109/TITS.2019.2909571.

[143] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, “DeepTP: An End-to-End Neural Network for Mobile Cellular Traffic Prediction,” *IEEE Netw.*, vol. 32, no. 6, pp. 108–115, 2018. [Online]. doi: 10.1109/MNET.2018.1800127.

[144] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, “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, 2019. [Online]. doi: 10.1109/JSAC.2019.2904363.

[145] M. Crawshaw, “Multi-task learning with deep neural networks: A survey,” *arXiv preprint*, pp. 1-43, 2020. [Online]. arXiv:2009.09796, 2020.

[146] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong and E. H. Chi, “Modeling task relationships in multi-task learning with multi-gate mixture-of-experts,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1930–1939, 2018. [Online]. doi: 10.1145/3219819.3220007.

[147] Y. Zhang and Q. Yang, “A Survey on Multi-Task Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 4347, no. c, pp. 1–20, 2021. [Online]. doi: 10.1109/TKDE.2021.3070203.

[148] X. Fang, G. Gong, G. Li, L. Chun, P. Peng and W. Li, “A general multi-source ensemble transfer learning framework integrate of LSTM-DANN and similarity metric for building energy prediction,” *Energy Build.*, vol. 252, p. 111435, 2021. [Online]. doi: 10.1016/j.enbuild.2021.111435.

[149] A. Maurer, M. Pontil, and B. Romera-Paredes, “Sparse coding for multitask and transfer learning,” *ICML 2013*, vol. 28, no. PART 2, pp. 1002–1010, 2013. [Online].

[150] M. Tiomoko *et al.*, “PCA-based Multi Task Learning : a Random Matrix Approach,” 2021. [Online]. HAL Id: hal-03420009

[151] W. Shao, Y. Peng, C. Zu, M. Wang and D. Zhang, “Hypergraph based multi-task feature selection for multimodal classification of Alzheimer’s disease,” *Comput. Med. Imaging Graph.*, vol. 80, p. 101663, 2020. [Online]. doi: 10.1016/j.compmedimag.2019.101663.

[152] T. Jebara, “Multitask sparsity via maximum entropy discrimination,” *J. Mach. Learn. Res.*, vol. 12, pp. 75–110, 2011.

[153] Y. Zhang, D. Y. Yeung and Q. Xu, “Probabilistic multi-task feature selection,” *NIPS 2010*, pp. 1–9, 2010.

[154] Caruana R. "Multitask learning,". 1997, vol.28, p.41-75, 1997.

[155] M. Suteu and Y. Guo, “Regularizing Deep Multi-Task Networks using Orthogonal Gradients,” *arXiv preprint* pp. 1–11, 2019. [Online]. Available: http://arxiv.org/abs/1912.06844.

[156] Y. Yang and T. M. Hospedales, “Trace norm regularised deep multi-task learning,” *ICLR 2017 - Work. Track Proc.*, no. 2014, pp. 2015–2018, 2019.

[157] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, “Cross-Stitch Networks for Multi-task Learning,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3994–4003, 2016. [Online]. doi: 10.1109/CVPR.2016.433.

[158] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, “Spatio-Temporal Wireless Traffic Prediction with Recurrent Neural Network,” *IEEE Wirel. Commun. Lett.*, vol. 7, no. 4, pp. 554–557, 2018. [Online]. doi: 10.1109/LWC.2018.2795605.

[159] C. W. Huang, C. T. Chiang, and Q. Li, “A study of deep learning networks on mobile traffic forecasting,” *IEEE Int. Symp. PIMRC*, pp. 1–6, 2018. [Online]. doi: 10.1109/PIMRC.2017.8292737.

[160] J. Zhang, Y. Zheng, J. Sun, and D. Qi, “Flow Prediction in Spatio-Temporal Networks Based on Multitask Deep Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, 2020. [Online]. doi: 10.1109/TKDE.2019.2891537.

[161] Q. Yuan, W. Zhang, C. Zhang, X. Geng, G. Cong and J. Han, “PRED: Periodic Region Detection for Mobility Modeling of Social Media Users,” *WSDM ’17*, pp. 263–272, 2017. [Online]. doi: 10.1145/3018661.3018680.

[162] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye, “Mining periodic behaviors for moving objects,” *16th KDD ’10*, p. 1099, 2010. [Online]. doi: 10.1145/1835804.1835942.

[163] T. M. T. Do and D. Gatica-Perez, “Where and what: Using smartphones to predict next locations and applications in daily life,” *Pervasive Mob. Comput.*, vol. 12, pp. 79–91, 2014. [Online]. doi: 10.1016/j.pmcj.2013.03.006.

[164] Q. Yuan, G. Cong, and Z. Ma, “Who, where, when and what: discover spatio-temporal topics for twitter users,” *Discov. Data Min.*, pp. 605–613, 2013. [Online]. doi: 10.1145/2487575.2487576.

[165] T. M. T. Do, O. Dousse, M. Miettinen and D. Gatica-Perez, “A probabilistic kernel method for human mobility prediction with smartphones,” *Pervasive Mob. Comput.*, vol. 20, pp. 13–28, 2015. [Online]. doi: 10.1016/j.pmcj.2014.09.001.

[166] J. Schloemann, H. S. Dhillon, and R. M. Buehrer, “A Tractable Metric for Evaluating Base Station Geometries in Cellular Network Localization,” *IEEE Wirel. Commun. Lett.*, vol. 5, no. 2, pp. 140–143, Apr. 2016. [Online]. doi: 10.1109/LWC.2015.2508935.

[167] Robusto and C. Carl. "The cosine-haversine formula." *The American Mathematical Monthly* pp.38-40, 1957.

[168] S. Kemp, “DIGITAL 2020: GLOBAL DIGITAL OVERVIEW,” Jan.2020. [Online]. https://datareportal.com/reports/digital-2020-global-digital-overview, accessed July 23,2021

[169] WalktheChat, “WeChat Impact Report 2018 shows impressive social impact - WalktheChat”, 2020, [Online]. Available: https://walkthechat.com/wechat-impact-report-2018-shows-impressive-social-impact/

[170] K. Kepaptsoglou, M. Asce, M. Karlaftis, D. Ph and M. Asce, “Multivariate Traffic Forecasting Technique Using Cell Transmission Model and SARIMA Model,” *J. Transp. Eng.*, vol. 8, pp. 174–182, 2015. [Online]. doi: 10.1061/(ASCE)0733-947X(2009)135.

[171] X. Zhou, Z. Zhao, R. Li, Y. Zhou and H. Zhang, “The predictability of cellular networks traffic,” *Isc. 2012*, pp. 973–978, 2012. [Online]. doi: 10.1109/ISCIT.2012.6381046.

[172] Y. Xu, F. Yin, W. Xu, J. Lin and S. Cui, “Wireless Traffic Prediction with Scalable Gaussian Process: Framework, Algorithms, and Verification,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291–1306, 2019. [Online]. doi: 10.1109/JSAC.2019.2904330.

[173] C. Zhang, P. Patras, and H. Haddadi, “Deep Learning in Mobile and Wireless Networking: A Survey,” *arXiv*, pp. 1–67, 2018.

[174] C. Zhang and P. Patras, “Long-term mobile traffic forecasting using deep Spatio-Temporal neural networks,” Proc. Int. Symp. Mob. Ad Hoc Netw. Comput., pp. 231–240, 2018

[175] X. Zhou, Z. Zhao, R. Li, Y. Zhou, and H. Zhang, “The predictability of cellular networks traffic,” 2012 Int. Symp. Commun. Inf. Technol. Isc. 2012, pp. 973–978, 2012

[176] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, “DeepTP: An End-to-End Neural Network for Mobile Cellular Traffic Prediction,” IEEE Netw., vol. 32, no. 6, pp. 108–115, 2018

[177] J. Wang et al., “Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach,” Proc. - IEEE INFOCOM, 2017

[178] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, “Citywide Cellular Traffic Prediction Based on Densely Connected Convolutional Neural Networks,” IEEE Commun. Lett., vol. 22, no. 8, pp. 1656–1659, 2018

[179] C. Zhang, M. Fiore, and P. Patras, “Multi-Service Mobile Traffic Forecasting via Convolutional Long Short-Term Memories.”

[180] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, “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, 2019

[181] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, “Spatio-Temporal Wireless Traffic Prediction with Recurrent Neural Network,” IEEE Wirel. Commun. Lett., vol. 7, no. 4, pp. 554–557, 2018

[182] C. W. Huang, C. T. Chiang, and Q. Li, “A study of deep learning networks on mobile traffic forecasting,” IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC, vol. 2017-Octob, pp. 1–6, 2018

[183] K. Zhang, Z. Liu, and L. Zheng, “Short-Term Prediction of Passenger Demand in Multi-Zone Level: Temporal Convolutional Neural Network with Multi-Task Learning,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 4, pp. 1480–1490, 2020

[184] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997

[185] M. Abadi et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” 2016, [Online]. Available: http://arxiv.org/abs/1603.04467.

[186] “GitHub - keras-team/keras: Deep Learning for humans.” https://github.com/keras-team/keras (accessed May 02, 2021).

[187] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012.