Shanghai Tourist Attractions Recommendation System

Yigao Xie, M.Sc.

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Future Networked Systems)

Supervisor: Meriel Huggard

August 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Yigao Xie

August 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Yigao Xie

August 19, 2022

Shanghai Tourist Attractions Recommendation System

Yigao Xie, Master of Science in Computer Science
University of Dublin, Trinity College, 2022

Supervisor: Meriel Huggard

With the current epidemic under control, Shanghai reopened a number of tourist attractions to visitors in June this year. If a tourist attraction is overcrowded, it can also affect the visitors' experience to a certain extent.

The project aims to design a system that will enable users to scan QR codes at tourist attractions to get real-time visitor counts, trends for the next two hours and recommendations for less-visited attractions. In this way, visitors can use it as an important reference in deciding whether to visit this tourist attraction.

The project uses real Shanghai visitor data to build an LSTM model for visitor number prediction, while the project uses Vue.js to build the front-end pages and Django to build the back-end code.

The system evaluation considered the performance of the LSTM model in terms of number prediction and in terms of the overall response time of the system. The preliminary work in this dissertation is promising and the system design is feasible and scalable for application in the tourist attractions mentioned in Shanghai.

Acknowledgments

I would like to express my gratitude to Prof. Meriel Huggard, Prof. James Ng for their valuable guidance and motivation during this dissertation. In particular, Prof. Meriel Huggard, who has provided me with continuous patient guidance and support at every stage of this dissertation.

I am grateful to my best friend Wenchao for his companionship, encouragement and inspiration.

Finally, I would like to thank my parents for providing me with the financial resources and enthusiasm to come and study at Trinity.

Yigao Xie

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter presents some background information on the project under study and briefly explains its significance and the potential impact it may have. The chapter also discusses why the project was developed and what the main objectives of the system are. Finally, the structure and organisation of the paper is explained.

1.1 Research background

The global spread of Covid-19 at the beginning of 2020 has had many impacts on the tourism industry everywhere. in 2020, China reported 2,879 billion visitor arrivals, a 52.1% decrease over the same period in 2018, and tourism income of 2.23 trillion yuan, a 61.0% decrease year-on-year. (Chengcai et al. (2022)). And due to the highly contagious and mutable nature of the virus and the mobility and congregation of people in tourism activities, the New Coronavirus epidemic has brought a huge impact and challenge to the global tourism industry Yang et al. (2021).

China's tourist growth may be separated into two stages by the end of 2021: the outbreak and the normalisation of epidemic prevention and control. (Chengcai et al. (2022)), with tourism always in a state of shock and recovery. During the outbreak phase, the tourism industry went through a process of flow control and capacity control to gradual relaxation and rapid recovery. to a gradual relaxation and rapid recovery.

The Shanghai Municipal Culture and Tourism Bureau announced on May 31 that after the middle and end of June, Shanghai would resume the opening of all kinds of cultural museums, performances, entertainment and other enclosed cultural and tourism venues in an orderly manner according to the phased arrangement of the city's full restoration of normal production and living order, as appropriate. Oriental Pearl Radio and TV Tower (outdoor section), Shanghai Wild Animal Park, Tinglin Park, Shanghai Grand View Park, Lu Xun, Park and Memorial Hall, Zhoupu Huahai Scenic Spot and 32 other tourist attractions will be the first to open their doors to visitors.

According to the big data from Where to Go (a Chinese hotel booking platform), hotel bookings in Shanghai rose 1.5 times on May 31 compared to May 30, and by 12 noon on June 1, hotel bookings in Shanghai had surpassed those of the same period on May 31, while searches for Shanghai scenic spots jumped twofold. Bookings for hotels in Shanghai were up 90% on the same period last month. Shanghai tourism is recovering fast.

Despite the current positive trend in the recovery of the Shanghai tourism market, the changing nature of the epidemic remains the most challenging factor for the market. Reducing crowds and controlling the number of visitors to tourist attractions will not only enhance the visitor experience, but will also go some way to controlling the outbreak.

The variability in the distribution of tourists in the time dimension is a familiar phenomenon to the tourism industry and academia. The phenomenon is well known to the tourism industry and academia. In terms of time scales, the temporal patterns of tourist flows are mainly can be measured in terms of long-term inter-annual variation, intra-annual seasonal/monthly variation, intra-weekly variation and intra-day variation. Intraday variation refers to the pattern of tourist intra-day variation refers to the differential distribution of visitors at different times of the day, and is the smallest of the time-scale measures of tourism flow research. It is the smallest unit in the time-scale measure of tourism flow research. Jia and Junyi (2016), based on the perspective of social media big data, investigates the intra-day distribution pattern of tourists in Xi'an, China, and finds that all enclosed scenic spots are typically 'daytime active', generally in a single-peak pattern, with tourist activity mainly concentrated during the day, with 11:00 a.m. being the peak time of tourist activity. This means that visitors usually choose a similar time of day. This means that visitors usually choose similar times to visit tourist attractions, which leads to a peak in visitor numbers at certain times of the day when the tourist attractions are open. If tourists concentrate on visiting the attraction during a certain time period, it will not only cause high pressure on the tourist attraction for epidemic prevention and control, but also very much affect the tourist experience.

Therefore, the aim of this paper is to develop a tourist attraction recommendation system based on visitor count prediction to provide a basis for decision making on whether to visit a particular tourist attraction or go to other less crowded tourist attractions when arriving at it, and also to balance the tourist imbalance between specific tourist attractions in Shanghai to some extent.

1.2 Research topic

The topic of this dissertation is the Shanghai Tourist Attractions Recommendation System, in which visitors can scan the QR code of each tourist attraction to get the current number of visitors to that attraction.

The system will also give a forecast of the number of visitors to the attraction in the next two hours, so that visitors can use the forecast line to determine whether they should continue to visit the attraction or go to another attraction. The system also gives a forecast of availability for the rest of the attractions and recommends the most available attractions to visitors. If the visitor subscribes to the attraction, an alert will be sent out when the number of people at the attraction drops below 50% to remind the visitor to visit.

1.3 Objectives

In order to implement the system proposed in 1.2, this study will achieve its objectives through the appropriate methods listed below.

- Collect real visitor data from several typical tourist attractions in Shanghai, and the corresponding weather data, and use them to build a model for each tourist attraction.
- Use the models to predict how the number of visitors to the tourist attraction will change over a 2 hour period.
- Based on the real-time travel time required to travel between each tourist attraction, predict the number of visitors to that attraction when they arrive at other tourist attractions. And the tourist attraction with the highest availability is recommended to the visitor.
- Writing web pages that can be accessed by scanning a QR code. The webpage includes weather data for the current tourist attraction, real-time visitor numbers, a trend line for the number of visitors over the next two hours, a list of tourist attractions calculated and recommended by the system and whether or not to subscribe to the button of the tourist attraction.

After the above objectives have been completed, this study will also evaluate the results of the project.

The desired outcome is the implementability of a system that meets these requirements and will improve the existing method of recommending tourist attractions.

1.4 Dissertation overview

The full dissertation is divided into six chapters, each with the following specific content.

- Chapter 1: Introduction - This chapter introduces the idea and concept of the project and discusses the motivation and purpose of this dissertation.
- Chapter 2: State of the art - This chapter provides a comprehensive overview of the most advanced technologies currently being used in the relevant field. It provides a detailed analysis of the techniques that can be used to collect information on visitor numbers in real time, short-term number of visitors forecasting methods, and number of visitors equilibrium models. In Section 2.2, this dissertation details current closely-related projects.
- Chapter 3: Design - In this chapter, the thesis presents a recommendation system based on specific tourist attractions in Shanghai, based on a summary of previous research, and describes its design details in detail.
- Chapter 4: Implementation - This chapter applies the design of the previous chapter to the step-by-step implementation of the system, and documents the outputs and results of each part of the system.
- Chapter 5: Evaluation - This chapter gives one of the quantitative evaluation methods that can be used to assess the results and performance of the system described above.
- Chapter 6: Conclusions and future work - This chapter summarises the dissertation report, highlighting the main contributions of this project to the current work and the limitations of the system, and suggesting areas for improvement and for further exploration in future research work.

Chapter 2

State of the Art

This chapter begins in 2.1 with a description of the different techniques used in this research, including the methods of a number of visitors data collection, short-term number of visitors forecasting methods and number of visitors equilibrium models. Afterwards, in 2.2, it continues with a discussion of some of the relevant aspects of the implementation and practice of these techniques that have emerged in the field. Finally, in 2.3, comparisons between them that are relevant to this dissertation are discussed.

2.1 Background

This section describes the techniques that will be used in this research and their interrelationships: The acquisition of the number of visitors data set will be the first step in this research. Once a certain amount of historical data is available, it will be modelled so that the model can be used to obtain the number of visitors forecasts for a short period of time. The system then uses a passenger flow equilibrium model to analyse and make recommendations based on the forecasts.

2.1.1 Number of visitors collection

Obtaining the current number of visitors to the attraction site from a hardware device is the basis for further modelling. The process typically uses a number of sensors or cameras to capture the original data, which is then uploaded to a backend server database using a microcontroller, before being processed for the next step of analysis and subsequently the number of people within the panoramic area. Depending on the type of data collected and how it is processed, the following main types of technology are currently available.

The first category uses conventional cameras or camcorders to capture images, which are then analysed based on computer vision. These methods use a specific camera to capture images or a camera to record a stream of image data and then use different analysis methods on these images to derive the number of heads passing by.

The most basic image processing count is target extraction using the values of RGB colours. Badr et al. (2021) has proposed a counting scheme based on this. The scheme takes its input from a stream of images captured by a surveillance video head, and then applies the same processing to individual image frames. For each image frame, the authors use OpenCV to perform the steps of removing the background, converting the RGB image to a binary image, performing Binary Large Object (BLOB) detection, analysing the BLOB result and determining the direction of movement of the object in sequence. Finally, the human count result is obtained by adding and subtracting.

Lian et al. (2021) designed a Dual Path Guided Detection Network (DPDNet) to improve crowd counting methods for RGB images. The DPDNet is comprised of a detection module guided by a density map and a detection module guided by depth. Specifically, to increase the performance of the dense/tiny head method based on detection, they suggest a module for density map-guided detection. This module uses the density map to classify heads and non-heads more accurately. Also, they take into account the possible size differences of the heads and therefore introduce a depth adaptive kernel to generate high-fidelity density maps, resulting in more robust regression of the density maps. The depth-guided detection module is used as a way to improve the ability to detect small heads. It is based on the principle of constructing a dynamic extended convolution in order to extract characteristics from heads of various sizes. It is based on the notion of constructing a dynamic extended convolution in order to extract characteristics from heads of various sizes.

Li et al. (2014) proposed a head detection-based headcount method which is able to count how many heads moved during the overhead camera recording. The approach comprises four primary components: foreground extraction, head detection, head tracking, and line crossing judgement. Starting from the first frame of the input video, the foreground is extracted using the VIBE technique (Barnich and Droogenbroeck (2011), Horn and Schunck (1981)) which has been tested in practice to be efficient, noise-resistant and have a low memory footprint. The region of interest (ROI) is then retrieved from the video using the foreground's binary pictures as a mask, i.e. the potential head. At this point, an Adaboost classifier based on LBP features that has been trained offline is used for the ROI to extract a more exact head. The study then provides a local head tracking approach based on the Meanshift algorithm and line crossing judgement for counting the number of passing heads in a video.

There are many other studies that have used such overhead cameras to obtain images and then perform headcount analysis, such as Teixeira and Savvides (2008), Segen (1996), Kim et al. (2003), which have to some extent optimised computer vision algorithms in order to to obtain a better headcount accuracy. The problem of occlusion caused by traditional side cameras (Yam et al. (2011), Zin et al. (2011), Liu et al. (2005)) has also been solved.

In a similar vein, Shahzad and Jalal (2021) used a similar processing process, but with infrared video. They perform foreground extraction of the infrared image by the Otsu technique and use the results to accurately identify pedestrians using template matching techniques in pattern recognition. In the pedestrian tracking step they used the Kalman filter and the Hungarian correlation method. This is then counted to give the number of pedestrians passing by in the video.

The second category is the use of depth sensors to collect data. Orrite-Uruñuela and Vicente-Dueñas (2018) offered a way to combine sensors and image analysis. To address the problem that the quality of images captured by conventional cameras can be affected by the position and brightness of the light source at the time of capture, the authors place an infrared depth sensor on top of the scene. The depth map generated by the camera is then obtained and combined with a machine learning model for image analysis to produce elapsed headcount data. In a similar vein, Coşkun et al. (2015) used a monochrome depth sensor, the Kinect, to collect depth data, which is then further used to draw a depth image based on the depth data using the Robot Operating System (ROS).

In addition to this, Dan et al. (2012), Fu et al. (2012) and others have combined camera pictures or camera images of the top view category with depth data in the hope of simultaneously solving the lighting problem posed by overhead cameras and the data loss problem potentially faced by depth cameras (Kim et al. (2010)), building on the strengths and avoiding the weaknesses to achieve better headcount results.

In alternative to computer vision methods, the third type of method uses only simple ultrasonic distance sensors, pressure sensors, or infrared sensors to count the number of people passing by. One such method, Hashimoto et al. (1997), has produced a fused multi-sensor headcount detector. The authors used an 8-element array of infrared detectors, pyroelectric PbTiO3 ceramics, an infrared transparent mirror and an oscillating mechanical chopper section to create an infrared ray detector. The difference between the measured and average output of the floor is then used to identify passing people in the area being detected. This tool is not only able to count the number of people passing by, but also to determine the direction in which people are walking.

There are other methods, such as the use of device-free passive RF. Depatla and Mostofi (2018) proposed a framework for Received Signal Strength Indicator (RSSI) based on WiFi links, which enables the counting of occupancy attributes within an area, such as the total number of people in the area. It is based on the principle that people can have a line-of-sight blocking effect and a multipath effect on a wireless link so that when a human body blocks the line of sight (LOS) link, the received signal strength is affected and weakened. Using this principle, the authors have carried out mathematical reasoning and experimental demonstration and applied the findings as theoretical support for the framework. It is able to measure the arrival rate of people in open areas as well as the headcount in closed areas and is the first joint passive estimation of crowd size in open and closed areas to be achieved using this technique.

2.1.2 Short-term number of visitors forecasting methods

Machine learning is the mainstay of short-term patronage prediction, and there are a large number of papers that have completed headcount predictions based on machine learning (Fu et al. (2016a), Tian and Pan (2015a), Sun et al. (2015), etc.). The categories of machine learning include supervised, unsupervised, semi-supervised, and reinforcement learning. Common techniques for supervised learning include regression analysis and statistical analysis. Common unsupervised learning algorithms are generative adversarial networks (GAN) and clustering. Of these, time series models in supervised learning are widely used and continually improved.

ARIMA model

Time is a necessary modelling variable for predicting changes in passenger flow; consequently, time series models in machine learning can be used. There are parametric and nonparametric approaches for analyzing time series. The Autoregressive Integrated Moving Average (ARIMA) model is an example of a parametric time series. It assumes that passenger flow conditions are a smooth process with a constant mean, standard deviation, and autocorrelation. A substantial number of traffic forecasting studies based on ARIMA have been undertaken in the field of traffic forecasting during the past few decades. (Smith et al. (2002), Williams and Hoel (2003), Chandra and Al-Deek (2009)).

An ARIMA model is constructed by changing a non-stationary time series into a stationary time series and then regressing the dependent variable on just its lagged values, as well as the present and lagged values of the random error term. Both autoregressive (AR) and moving average (MA) models are considered in ARIMA models, also written as ARIMA (p,d,q). Where p stands for the order of the autoregressive model, q stands for the order of the moving parallel model, and d determines the number of orders of differencing.

Song and Zhou (2019) modelled the flow of visitors to all 3A and above picturesque locations in Shanghai on holidays and weekdays from 10 am to 3 pm using an ARIMA model. The authors argue, based on their model, that the visitor flow to Shanghai's 3A and above every places exhibits varied volatilities and interacts with time.

Neural networks and the Kalman filter are non-parametric methods, which means that the structure and parameters of the model are not fixed.

Artificial neural network

According to Ma et al. (2015), in the subject of traffic flow forecasting, prediction algorithms have gradually switched from classical statistical models to artificial intelligence (CI). Flexible and more inclusive of outliers, missing data, and noise (Karlaftis and Vlahogianni (2011)). Artificial neural networks (ANNs), the most prominent of these, have also found a place in traffic forecasting.

Using neural networks to estimate traffic flow as early as 1994, Hua and Faghri (1994) On the basis of this discovery, scientists have created a range of neural network forecasting approaches. For instance, feedforward neural networks (Park and Rilett (1999)), radial basis frequency neural networks (Park et al. (1999)), neuro-fuzzy neural networks (Yin et al. (2002)), and recurrent neural networks (Lingras et al. (2002)).

To further enhance the performance of the model, Hochreiter and Schmidhuber (1997) invented the Long Short Term Memory Recurrent Neural Network (LSTM) framework, which solves the long-term dependency problem of RNNs by introducing the gate mechanism and achieves good prediction results after training the model with sensor data. For short-term passenger traffic prediction (Tian and Pan (2015b)), the LSTM RNN outperformed other models, such as random walk (RW), support vector machine (SVM), stacked auto-encoder (SAE), and FFNN.

In contrast, Fu et al. (2016b)'s research discovered that while the LSTM model was highly predictive for short-term passenger flow prediction, it was challenging to construct the hidden layers, learning rate, and number of network iterations. As a result, several enhanced LSTM models have been developed in the years since.

PENG et al. (2020) adds a fully connected neural network layer before the input layer of the LSTM and sets its initial weights to be identical to the beginning weights of the LSTM. With this configuration, the network's depth is raised, which in turn enhances the network's feature extraction efficiency. In order to prevent overfitting during training due to too many layers, the authors ran multiple network search experiments and concluded that the optimum fitting effect could be reached by adding three more layers. Using the entrance/exit data of Beijing West Station as the study object, the short-term passenger flow prediction model of the augmented LSTM network is more precise than the traditional LSTM network.

Jing et al. (2021) improves the LSTM through the classification of statistical features on time series. The authors categorise all time series and non-time series features, maintaining the consistency of the time axis of time series features in the LSTM input. This method can reduce the LSTM network's complexity and increase the model's computational efficiency. In terms of model design, the authors partition the features into three input components utilising three channels. Temporal features 1 and 2 are input from the left and right channels, and after learning and processing through the LSTM recurrent layer, they are input into the Dense layer containing the fully connected neural network for further learning and processing; Non-time series features are not required to traverse the LSTM recurrent layer; rather, they begin learning and processing straight via the Dense layer.

Kalman filter

The essence of the Kalman filter is to reconstruct the state vector of a system from quantiles, and it provides a computationally efficient (recursive) approach to predict the state of a process so as to minimise the mean squared error. It is remarkable for its capacity to estimate the state of a system at three different times, past, present, and future, from a single model, despite the fact that the real nature of the system being modelled is uncertain (Kalman (1960)).

Based on the Extended Kalman Filter (EKF), an increasing number of studies focus on traffic flow research. Based on this, Wang and Papageorgiou (2005) presents a general approach for estimating the total traffic state of a highway segment in real time. The limiting EKF (LimEKF) and the traceless Kalman filter were proposed by Antoniou et al. (2007). And the conclusions are applied to the road system in Southampton, United Kingdom. For forecasting travel times, Van Lint (2006) suggested a delayed EKF method for gradual online training of a data-driven trip time prediction model: SSNN. In the field of footfall prediction, no pertinent research has been uncovered.

Gray models

Gray model (GM) is a gray differential prediction model that gives an imperfect, long-term description of the development pattern of objects based on insufficient or restricted data.

Based on Gray system theory, Lingbin et al. (2009) created gray models for the quantity of international tourists in China. in different time domains based on the features of dynamic change and a realistic analysis. Simultaneously, studies and forecasts are conducted, and the resulting elevation fitting precision is high in comparison to actual values. In addition, the authors note that short-term series models may result in huge long-term forecasting mistakes and that long-term series may contain a great deal of data, but their accuracy is not always good.

However, the gray model lacks self-learning, self-organizing, and self-adaptive characteristics and is ineffective in processing nonlinear data (Yuan et al. (2009)). In these regards, the neural network model can supplement the GM. Consequently, numerous studies based on models of gray neural networks have arisen.

Al-Deek (2002) proposed a method combining BP neural network and time series analysis to estimate the flow of heavy trucks arriving and departing the port, achieving a relatively accurate forecast.

Li et al. (2021) examined the change in China's Shandong Province's visitor population. Among these, the correlation analysis method selected variables influencing the tourist population, such as tourist consumption level, construction of tourism facilities, national macro policies, and weather conditions, and built a GM(1,1) model based on these variables. The authors then constructed a combined GM-BP neural network model to accurately forecast the number of tourists visiting Shandong Province.

2.1.3 Number of visitors equilibrium models

There are two traditional traffic allocation models: user equilibrium (UE) and system optimization (SO) (Wardrop (1952)). They adhere to Wardrop's first and second principles, respectively, and the UE model is based on two assumptions: that road users have comprehensive information about alternative network paths, and that they can use any path if the currently used path is congested. This corresponds to a visitor equilibrium for tourist attractions, as tourists have access to comprehensive information about all attractions and can choose an alternative attraction if the current one is too crowded.

In order to employ UE for prediction in actual scenarios, Daganzo and Sheffi (1977) presented a stochastic user model (SUE), as the authors contend that no road user may unilaterally reduce their perceived trip costs by changing their path. Based on this, Jahn et al. (2005) introduced the average length of pathways in order to develop a system optimisation approach that accommodates specific requirements. Jiang et al. (2012) proposed a model that accounts for some users' distance restrictions (e.g. e-bikes). Liu et al. (2018) offered a stochastic -path logit user balance (NPSUE). This model considers not only the perceived faults of road users in path selection, but also the heterogeneity of users, so that the number of finite sets of paths varies for each user. In addition, the authors determined that NPSUE had superior performance over UE, SUE, and SO.

Regarding tourist attractions, there is no documented and systematic equilibrium model for travelers. Nonetheless, if road traffic is considered, the aforementioned equilibrium model for road traffic may serve as a useful reference.

2.2 Closely-related projects

There have been a number of projects in the field of tourist attraction recommendation systems. On location-based social networks, Cho et al. (2011) and Gao et al. (2013) primarily utilize location information data based on the diverse search interests of members (LBSN). Both Yoon et al. (2010) and Zheng and Xie (2011) produce recommended destinations based on user-generated GPS tracks. Zhang and Chow (2015) and Yuan et al. (2013) take into account that the influence of time on the recommendation system is necessary.

Wang et al. (2016) takes into consideration the actual crowding of tourist attractions. The authors propose a new algorithm, Personalised Crowd-Aware Travel Recommendation (PersCT), to recommend personalized routes while avoiding as much as possible the busiest times at tourist destinations. The authors represent the travel recommendation problem as a directed problem with multiple constraints and add the extracted user interests to the constraints; the PersCT algorithm is based on ant colony optimisation and is combined with user interests and the popularity and crowding of tourist attractions. Additionally, the algorithm is assessed using pedestrian traffic data from a real-world sensor data set and user trip history which is gathered from a Flickr photo data set.

Migliorini et al. (2021) provides a solution that takes into account both the current and future crowding levels of tourist attractions. The authors propose this approach to effectively explore the solution space using a MapReduce implementation of a multiobject optimization problem and to balance users between POIs by factoring in projected crowding levels. In this case, the distance from the user's current location to the POIs, the time the trip is likely to last, the mode of travel, and the smoothness of the trip are all taken into account as system parameters. Further, they used real data sets to evaluate the system and added the consideration of paths where visitors showed great subjective interest to the recommendation system.

2.3 Comparison between research

Technology is constantly evolving and changing, and researchers are working to apply newly developed technologies to various fields. Some are broad and generic, while others specialise in a specific case. It is therefore valuable to compare the different technologies across the board. In conjunction with the analysis and presentations in the previous sections, the research and papers in several areas covered in this paper are compared and summarised below.

2.3.1 Comparison between methods of collecting visitor numbers

Much of the research on the collection of visitor numbers is based in the field of image analysis, where the difference is mainly in the way the image data is captured. In the field of image analysis, there is a very large body of research that is constantly improving image analysis algorithms, resulting in breakthroughs in recognition accuracy and recognition efficiency time and time again. However, the hardware set-up and maintenance costs of this approach are generally high. And its main data is images, which leads to a situation where the speed of data transmission may be affected during practical applications.

In contrast, an approach that simply transmits simple sensor values may be more suitable for monitoring the number of people in small areas, especially when the system uses the MQTT protocol popular in IoT system applications. The actual deployment cost may also vary significantly due to the accuracy, complexity and cost of the selected sensors. Often there is not just one sensor at a sensing node, but multiple sensors working in conjunction with each other, making this approach a relatively complex system architecture in terms of data acquisition.

Other approaches, on the other hand, offer a variety of new research ideas for subsequent researchers. They each have their own unique advantages and disadvantages and the most suitable application scenarios.

2.3.2 Comparison between short-term visitor forecasting models

This area is perfectly suited to the application of time series models, and time series models tend to perform well on data sets. The most basic ARIMA model is more selective about the data set and not all data sets can be used with the ARIMA model and get a high level of accuracy in prediction results. Therefore, this model is more suitable for scenarios where the accuracy of the predictions is not as high as required.

The LSTM is another time series model based on the RNN model, which has a large improvement in generalizability, accuracy and performance over the ARIMA model and the underlying RNN model (Tian and Pan (2015b)). And the LSTM has problems that are negligible for scenarios where the accuracy of prediction is not so demanding. Subsequent researchers have continued to improve on it. Changing the structure of the neural network and increasing the complexity of the network is one of the main improvement directions. However, it is somewhat specific, that is, it can greatly improve performance for specified application scenarios, but its generalisability for other application scenarios still needs to be tested.

In addition to the above methods, the Kalman filter and the gray model have also been used in many studies. The Kalman filter method is not well suited to predicting the number of visitors, while the gray model is weaker in handling non-linear information. Both methods have obvious differences in principle from the time series model and no research has yet been conducted to make experiments on the differences in performance. However, the gray model has already had some applications in the field of population movement prediction, so an attempt could be made to improve the gray model with specificity and then apply it to a tourist attraction recommendation system.

2.3.3 Comparison between equilibrium models of visitor numbers

The current applications of traffic balancing models and algorithms are mainly focused on the field of road traffic planning. Several models on UE mentioned in section 2.1.3 have not yet been applied to research in tourist attraction recommendation systems. They may be a new direction for researchers to explore in the next step, but will also require a lot of padding for further research. In terms of the logic and characteristics of the models themselves, the NPSUE model may be more suitable for porting to tourist attraction recommendation systems than the base UE model because of the specificity of the user in choosing roads that it takes into account.

2.3.4 Comparison between closely-related projects

The customisation of recommended tourist attractions according to the individual search needs of different users has been a major direction in recent years for tourist attraction recommendation systems. The most basic is based on GPS, which has the advantage of being a simple and efficient algorithm, almost tailored to the user's current location, which helps the user filter the most favourable tourist attractions in terms of commuting time to the maximum extent. Further research has added to this the user's behavioural trajectory, which allows the recommendations to be tailored to a certain extent to the user's personal interests. The disadvantage of this category is that it only targets the geographical distribution of tourist attractions and does not take full advantage of the huge amount of information available on the tourist attractions themselves.

The Wang et al. (2016) study, on the other hand, to a certain extent due to the consideration of the crowdedness of the attraction, thus making the recommendation results further personalised to the user's needs. And the PersCT algorithm proposed in this paper has not yet been published by any new researcher giving it an improved study.

The solution proposed by Migliorini et al. (2021) does not address the level of congestion at the tourist attraction itself, but rather the level of congestion on this leg of the journey from the user's current location to the tourist attraction. Although this research is more in the realm of road traffic, its integration of road information into a tourist attraction recommendation system is very worthwhile. The approach proposed by the project in this dissertation to take into account the number of visitors to tourist attractions within the recommendation algorithm also gains inspiration from the above research.

2.4 Summary

This chapter provides an overview of the papers and research relevant to this project.

In section 2.1, three separate aspects of background are reviewed. They are number of visitors data collection, short-term number of visitors forecasting methods and number of visitors equilibrium models. and in each of these aspects the relevant techniques and related research. In section 2.2, different tourist attraction recommendation systems that are closely related to the research project of this thesis are reviewed. In section 2.3, a brief comparative analysis of the studies mentioned in sections 2.1 and 2.3 is presented.

In the next chapter, each component of this project will be designed based on the content of this chapter.

Chapter 3

Design

Based on a review of previous relevant research, this chapter will discuss various possible design options and the theoretical basis for the system design, based on the requirements of the project, prior to its actual implementation.

First, a description of the overall project architecture is given in section 3.1, followed by detailed design ideas for each component of the project in the next few sections.

In section 3.2 the design of the data collection process is presented, including the types of data to be collected, the data collection tools and the collection methods.

In section 3.3 the visitor number forecasting model is presented.

In section 3.4 the recommendation strategy to be used by the system in the project is presented.

Finally, the methodology for the development of the system application is presented in section 3.5.

3.1 Overview of the approach

As shown in Figure 3.1, the architecture of this system will be divided into five layers.

In the physical layer, I use ESP32 as the microcontroller and use two sonar sensors to monitor the number of visitors entering and exiting the tourist attraction to get a real-time count of the number of visitors actually being in the park.

In the foundation layer, I used mySQL as the database and Tencent Cloud as the cloud server. This ensures that the system's user interface can be accessed from any region at any time, and also supports multiple users accessing at the same time.

In the data layer, my data mainly consisted of weather data and the number of current visitors.

In the service layer, I use Django to build the backend.

In the application layer, I use Vue.js to design the front-end pages.

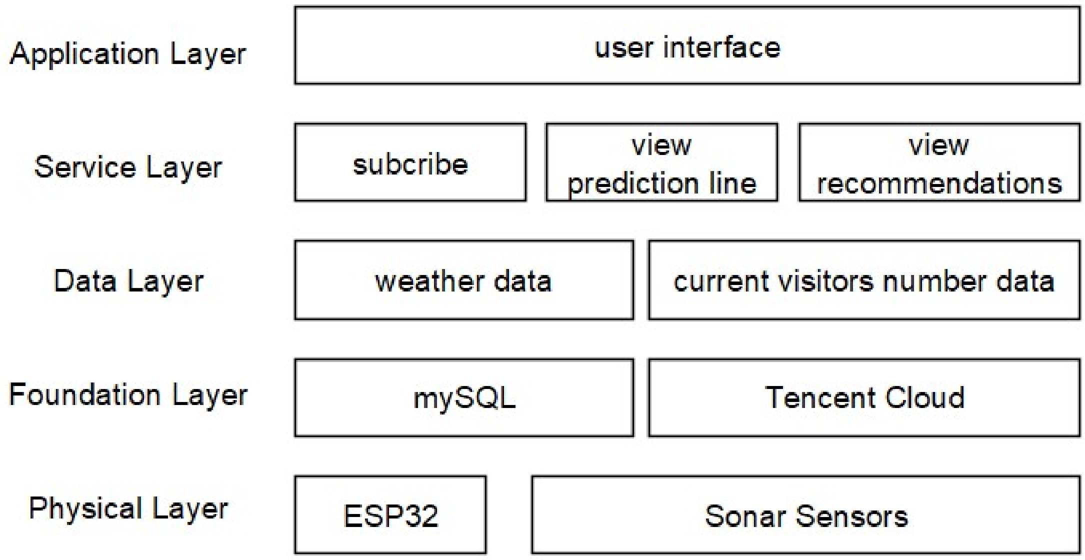


Figure 3.1: The overall layered structure of the project

3.2 Data set collection

In order to construct a suitable data set for this project, in section 2.1.1, I have studied and understood in depth the current methods commonly used to collect real-time data, especially for visitor numbers are detection methods. The main data for visitor number prediction lies in the number of people and environmental factors that may affect visitor arrivals. However, as the data set required for this project is very specific to just the number of visitors to Shanghai's tourist attractions and the environmental data of Shanghai, there is no public data set available for research and analysis. This led to the first step in my research being to obtain a suitable data set.

3.2.1 Requirements of data set

The purpose of constructing the data set is to train a model to predict the number of visitors over time in the future. The required data set should therefore include real-time visitor numbers for each tourist attraction in Shanghai that is included in the system and the environmental factors that have an impact on visitor arrival behaviour. The main common environmental information that can be collected is the weather. In addition, as the model should be a time-dependent sequential model, each data set should include time point information. Also, reasonable time intervals should be considered. As the number of visitors to tourist attractions can change relatively quickly when visitors are coming in and out more frequently, the time interval should not be too long for the accuracy of the prediction.

Based on the above discussion, the data set should be collected to meet the following requirements.

- The data set should include continuous temporal data with an interval of 1 minute.
- Each data entry should include the current number of visitors to the tourist attraction.
- Each data entry should include environmental data based on weather information, specifically wind speed, pm2.5 index, precipitation and temperature.
- All data should cover at least one full natural day.

3.2.2 Data collection method

The project's approach to data collection at the physical layer is to use the ESP32 as a microcontroller and four ultrasonic sensors HC-SR04 to detect changes in the number of visitors to tourist attractions.

The MCU chip in the ESP32 hardware platform is a device module that can be used as a stand-alone running application and its main carrier can provide WiFi and Bluetooth functionality via SPI/SDIO or I2C/UART interfaces. In addition the ESP32 module requires very few peripheral devices for safe and reliable data communication and processing functions. This lightweight microcontroller has therefore been chosen as the controller to quickly and easily collect the information collected by the sensor and send it to a remote database.

The ultrasonic sensor HC-SR04 operates on the concept that the transmitting module Trig transmits ultrasonic waves in a certain direction and begins timing at the point of emission. The ultrasonic waves travel through the air and return quickly when they contact an obstruction, and the ultrasonic receiver Echo ceases timing as soon as it receives the reflected wave in order to calculate the time T. The module then operates according to the distance calculation formula, which results in the distance to the target obstacle.

This is because the speed of propagation of ultrasound waves in air is influenced by temperature. The speed of propagation V of the sound wave in air is related to the temperature t by

$$V = 332 + 0.607t(m/s)$$

Since in this project the distance change is only used to detect if a visitor is passing by, the requirements for accuracy are not high, so the speed of sound at 20 degrees Celsius can be taken for calculation, at which point the distance calculation formula is as follow:

$$L = 344M/S \times T \div 2 = 172M/S \times T$$

The exact installation design of the hardware equipment is shown in Figure 3.2. Two of the sonar sensors are used to monitor the number of visitors entering the attraction, they are designed to be at different heights and have a front-to-back relationship and are spatially staggered. The other two are used to monitor the number of visitors leaving the attraction. These two are subtracted and the result is the number of visitors within the attraction.

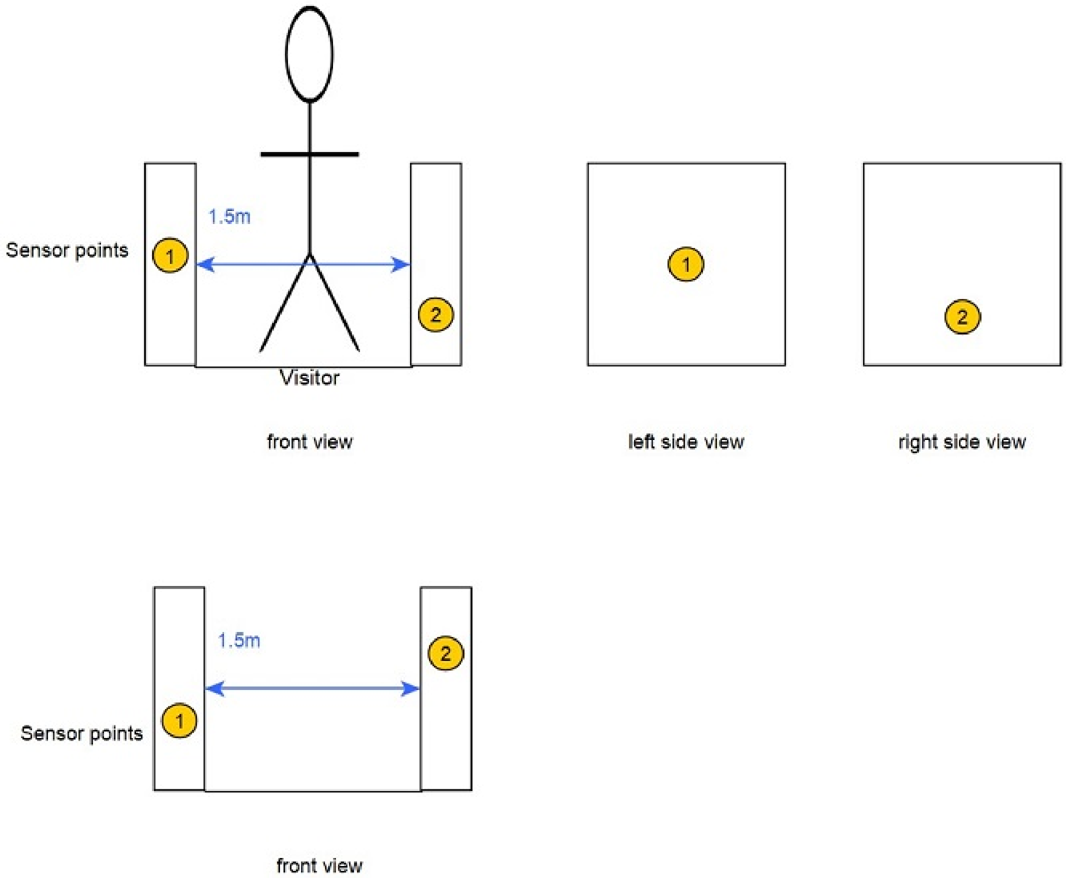


Figure 3.2: Diagram showing the location of the ultrasonic sensor

The distance originally measured by the ultrasonic sensors should be 1.5 metres, if the distance changes abruptly to any value less than 1.5 metres then someone may have passed by. To rule out anomalies, if two ultrasonic sensors detect a sudden change in distance in sequence, the site will be deemed to be occupied and the number of visitors will be increased or decreased accordingly.

This device should be installed at each tourist attraction depending on the actual number of entrances and exits, so that information on the number of visitors to the attraction can be collected.

3.2.3 Access to simulation data

Due to geographical constraints, I did not have the means to place the hardware device in a specific tourist attraction in Shanghai, so I found some publicly available real data from the web to form the data set used to train the model.

- Visitor count data:

  Real-time visitor data for all tourist attractions in Shanghai was obtained from the Real-Time Visitor Count Portal for scenic spots published by the Shanghai government, which can be found at https://lysh.smgtech.net/scenicArea.html. This is shown in Figure 3.3. This website is updated in real time, so python can be used to read the number of visitors, time and date of the desired tourist attraction in real time, and also to obtain the maximum number of visitors that each tourist attraction can accommodate.

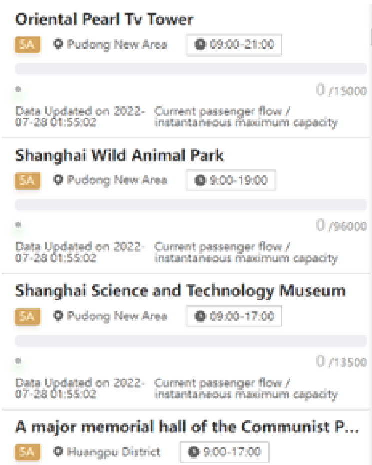


Figure 3.3: Visitor data for tourist attractions made public by the Shanghai government

- Environmental data:

  It is sourced from the Shanghai Central Weather Station's weather data distribution website at http://sh.weather.com.cn/index.shtml. This is shown in Figure 3.4. This website provides the wind speed, rainfall, pm2.5 values and temperature data needed for the data set, which can also be obtained in real time using a web crawler and saved to a database.

Figure 3.4: Weather data published by Shanghai Central Weather Bureau

3.3 Model selection

With a data set in hand, the next step is to consider what model to choose to train and predict on the data set.

This thesis reviews the more typical methods for predicting short-term visitor numbers in current research in section 2.1.2. Of these, the LSTM model is the most commonly used method, and with minor refinements, it is usually able to demonstrate a more than satisfactory level of accuracy.

The LSTM model is a type of deep learning that solves the RNN short-term memory problem by adding Gates to the RNN model, allowing the recurrent neural network to really make effective use of long-range temporal information. the original LSTM architecture is shown in Figure 3.5.

in 2000, Gers, Schmidhuber, and Cummins modified the original LSTM. They introduced a forget gate to the original architecture and defined the new LSTM architecture as shown in Figure 3.6. (Yu et al. (2019)) At this point, the LSTM mathematical expression is as follows:

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f)$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i)$$

$$\tilde{c}_t = tanh(W_{\tilde{c}h}h_{t-1} + W_{\tilde{c}x}x_t + b_{\tilde{c}})$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o)$$

$$h_t = o_t \cdot tanh(c_t)$$

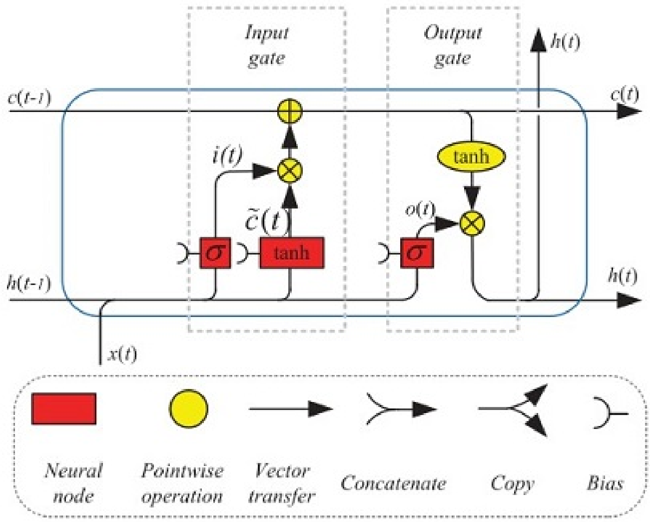


Figure 3.5: Original LSTM architecture(The source of the figure is Yu et al. (2019))

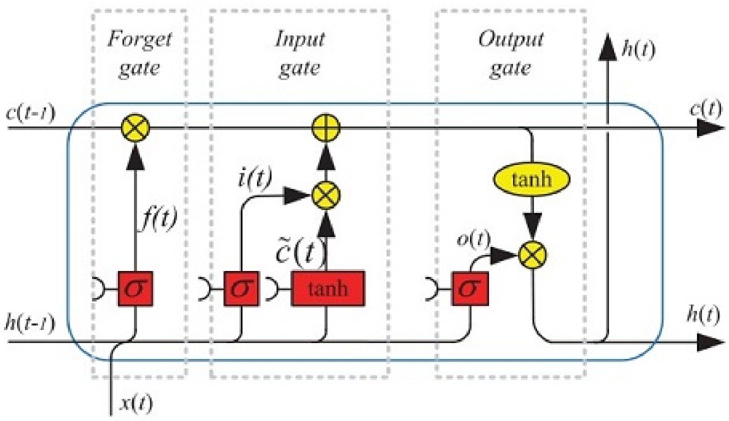


Figure 3.6: Architecture of LSTM with a forget gate(The source of the figure is Yu et al. (2019))

This project uses the LSTM model provided by tensorflow.keras.layers. The data collected in the previous section was pre-processed with data and put into it for training to get a model for each specific tourist attraction separately and then saved as h5 files.

3.4 Recommendation rules

Once a visitor has scanned the QR code of a tourist attraction, they may or may not choose to visit the attraction. The system should give recommendations regardless of whether the visitor visits or not.

The recommendation of tourist attractions in this system is also a problem of visitor balancing between each selected tourist attraction. In this dissertation, some existing methods for balancing the number of visitors are reviewed in section 2.1.3. Most of these methods are applied to traffic flow equalisation in the traffic domain and are not very suitable for visitor equalisation at tourist attractions. Therefore some special recommendation strategies should be investigated for the specific needs of this system.

As a general rule, tourists will tend to visit tourist attractions with fewer people. However, given that the maximum number of visitors that each tourist attraction can accommodate varies, it is not reasonable to simply count the number of visitors in a tourist attraction to determine which tourist attraction to recommend. The difference between the number of visitors available in a tourist attraction and the total capacity, i.e. the availability of the tourist attraction, should be calculated. Availability reaches a minimum value of 0 when the number of visitors is full, and a maximum value of 1 when there are no visitors.

Therefore, the recommendation rule for this system is as follows: the highest availability of the remaining tourist attractions when visitors arrive is recommended.

For example, if the current tourist attraction is A, then the recommendation process is as follows:

1. Calculate the distance from A to another tourist attraction, B, and estimate the time t it will take for the tourist to reach B.

2. Calculate the number of visitors in the tourist attraction after t has elapsed since the current time for B.

3. Combine this with the maximum number of visitors that B itself can accommodate and calculate the availability of B at this time.

4. Perform this calculation for each of the tourist attractions.

5. Compare the availability of B and the remaining tourist attractions separately and rank them.

6. Recommend the top 3 tourist attractions with the highest availability to visitors.

3.5 Application design

Every visitor can scan the QR code of a tourist attraction on their mobile phone to get data on the number of visitors to the tourist attraction and information on recommendations. The application does not require a user login or registration and therefore does not need to store any personal information of the visitor. Once the QR code has been scanned, a web page will be accessed. From the web page they can access the current number of visitors to the attraction, a predicted line chart of the number of visitors for the next two hours and the tourist attractions recommended by the system. They can also choose whether or not to subscribe to the attraction. If they subscribe to the attraction, they will be alerted by a pop-up on the page when the system monitors that the attraction has less than 50% of the maximum number of visitors in real time. If not subscribed, this page will also continue to update the data over time.

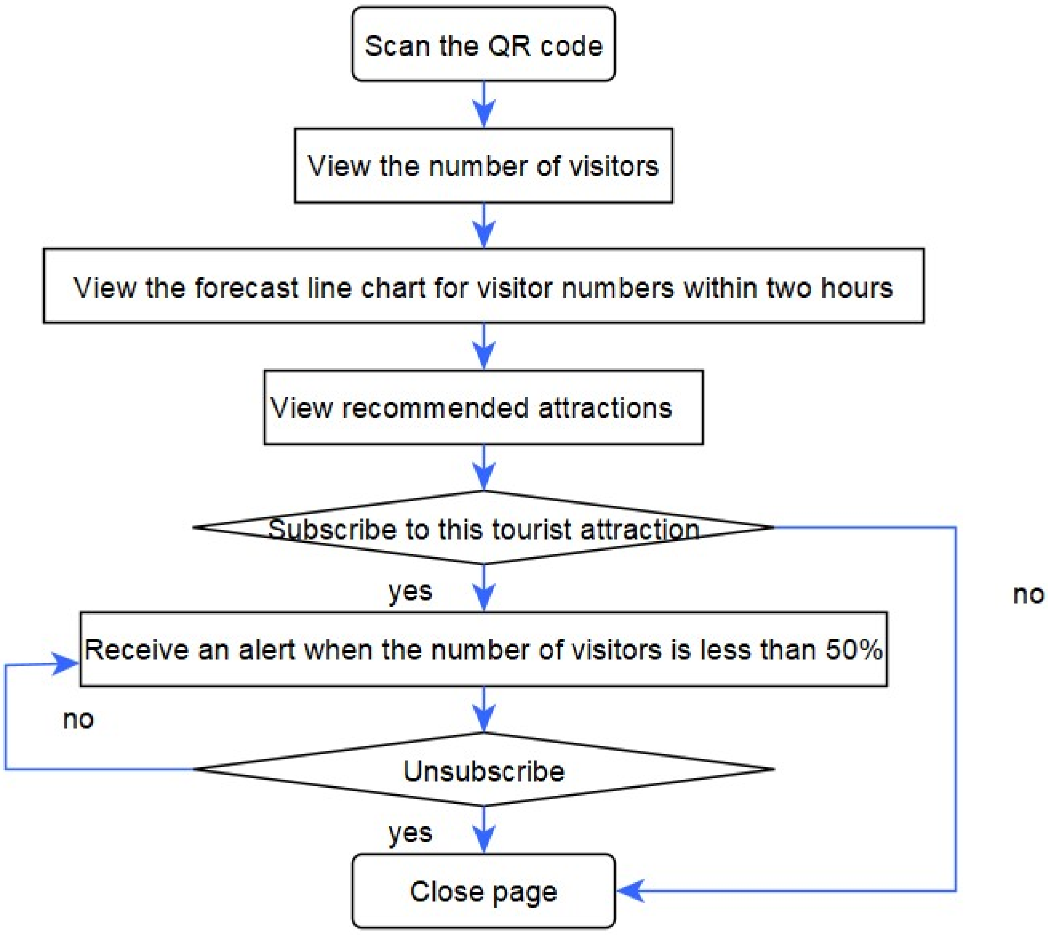


Figure 3.7: Flowchart of the application's user use

The above process of visitor use of the application is summarised in the User flow, as shown in Figure 3.7

3.6 Summary

This chapter describes the design scheme planned for use in this project.

In section 3.1, it is described that the system will be divided bottom-up into an entity layer, a base layer, a data layer, a service layer and an application layer. In section 3.2, the requirements for the data set and how it will be collected are given. One of the two main ways of collection is the use of ESP32 and ultrasonic sensors, and the second way is the use of publicly available official data. In section 3.3, it is described that the model that the system will use is the LSTM model. In section 3.4, the recommended rules that will be used by the system are specified in detail. In section 3.5, a user flow diagram of the web application is given and the content that should be included on the page is presented.

In the next chapter, this dissertation will report on the results of the implementation of the above design.

Chapter 4

Implementation

Having completed the overall design of the system, the next step was to take the planning through to coding. Firstly section 4.1 describes the framework and technology used for the project as a whole, and then in the subsequent sections the output process and results of each part of the system are presented separately.

4.1 Frameworks and technologies

This project uses a separate programming approach for the front and back ends. The front-end pages are written using Vue.Js, the back-end support code is written using Django framework, the communication protocol between the front and back-end is HTTP protocol, the programming language is Python, and the cloud server is Tencent Cloud.

4.1.1 Frontend

The system's front-end pages are developed using Vue.js, a platform for developing user interfaces that is progressive. Vue, unlike other heavyweight frameworks, is built for gradual development from the ground up; the basic Vue library focuses just on the view layer and is extremely easy to combine with other libraries or existing applications. In contrast, Vue is fully capable of powering sophisticated single-page apps built using single-file components and libraries supported by the Vue ecosystem.

4.1.2 Backend

The system's back-end services enable the Django framework to be used for programming.Django is a Python-based web framework for the rapid creation of web applications. It is able to avoid the installation and dependency issues found in other frameworks, such as Flask where many plugins need to be installed by the user themselves, sometimes with unknown errors. django is based on the Model View Template (MVT) architecture, whose core operations are Create, Retrieve, Update, Delete (CRUD). Figure 4.1 below shows a diagram of how a Django application interacts with the system structure.

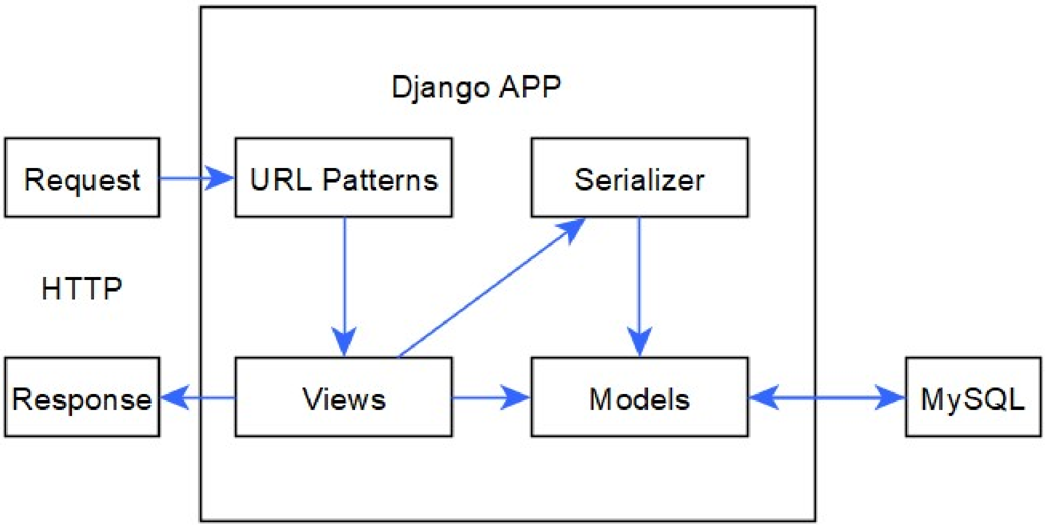


Figure 4.1: Django application interacts with the system structure

4.1.3 Cloud server

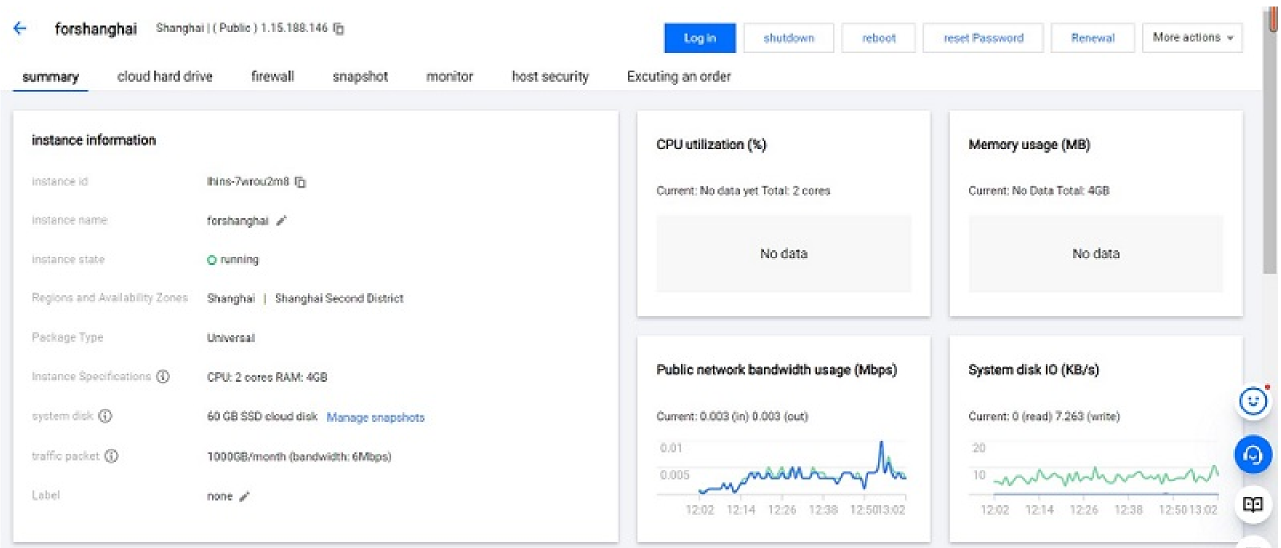


Figure 4.2: Running CVM

The project needed to be run on a cloud-based server as visitors from different locations needed to be able to access the site by scanning the code. I chose Tencent Cloud's

(Cloud Virtual Machine, CVM). It provides a scalable computing service. Compared to traditional servers, building the project on a CVM using a cloud server avoids the need to estimate resource usage and upfront investment. Figure 4.2 below shows the details of my CVM instance and some data when it was only running this project.

4.2 Data preparation and modelling

The project is based on two ways of obtaining data. In practice, a hardware device should be used to read the number of visitors to a tourist attraction in real time, and due to the limitations of my current location, I took a web crawl to obtain the real time visitor count data to use it as the raw data set for modelling.

4.2.1 Data collection

The data collection is divided into two parts, one is the hardware implementation which records the changes of visitors through changes in sensor values. However, due to geographical limitations, the hardware is not able to collect real visitor data. Therefore, in order to compensate for this, the second part is the crawling of real data via the web.

Hardware

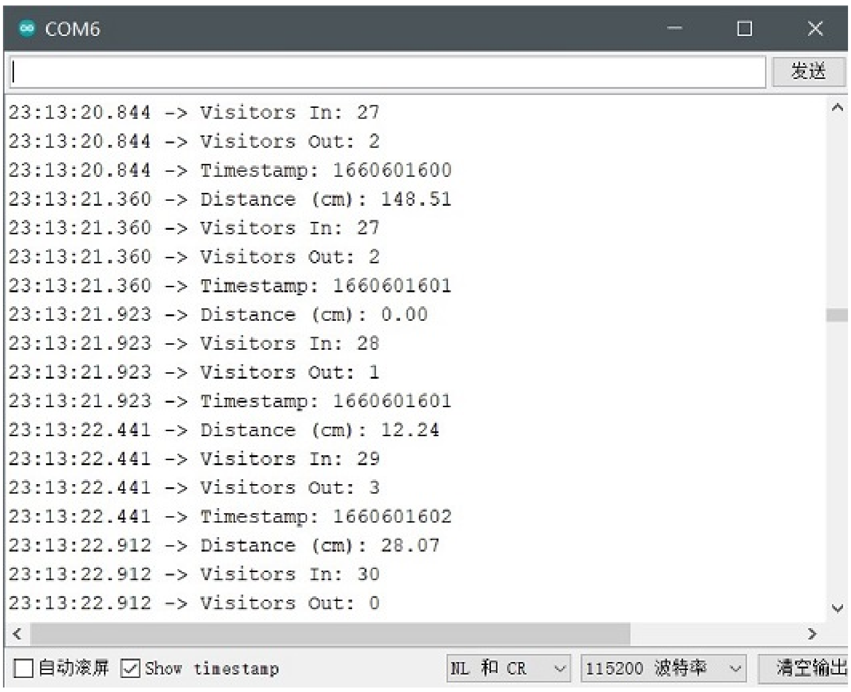


Figure 4.3: Information displayed by the Arduino serial monitor

The project uses Arduino to write a C program that is burned into the ESP32 to control the ultrasonic sensor SR04 to detect changes in distance at a specific location. When the distance value is less than 50 cm, it indicates that a visitor is passing by.

Figure 4.3 shows the data from the Arduino serial monitor after debugging the ESP32 and the ultrasonic sensor. As can be seen from this graph, when the distance was 148.51 cm, the number of entering visitors was 27. At the three subsequent time points, the distance was 0.00 cm, 12.24 cm and 28.07 cm in that order, all less than the set boundary value of 50 cm, indicating that three consecutive people passed by. The count of entering visitors therefore increased three times in succession, from 27 to 30. and the number of visitors going out was a random value.

Web crawling data

	A	B	C	D	E	F	G	H	I	J	K
1	time	A	B	上海科技	中共一大	上海寿安	上海东禾	上海亭林	上海崇明	上海泰会	中共二大
2	02:58.9	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
3	03:58.5	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
4	04:58.0	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
5	05:57.5	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
6	06:57.1	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
7	07:56.6	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
8	08:56.2	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
9	09:56.0	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
10	10:55.5	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
11	11:55.0	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
12	12:54.6	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
13	13:54.1	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
14	14:53.6	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200
15	15:53.2	0/15000	0/96000	0/13500	0/1875	0/5000	0/1500	0/3500	0/375	0/1000	0/200

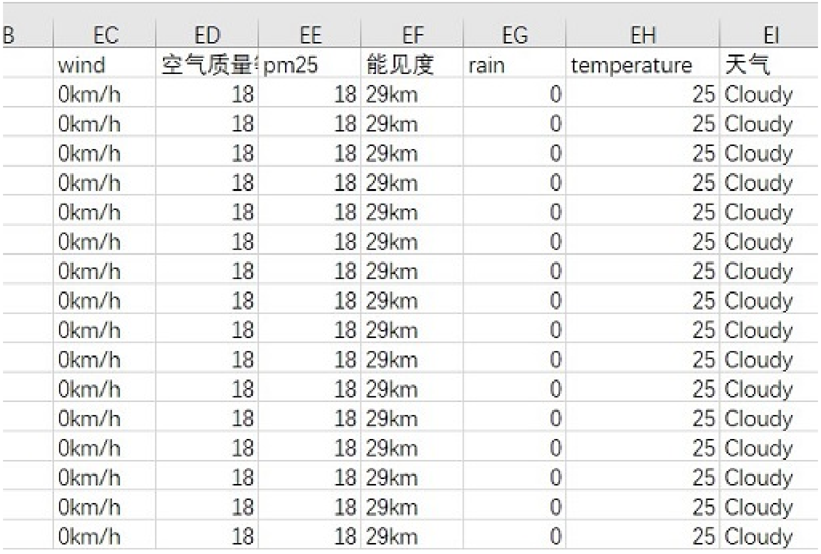
Figure 4.4: Example of a partial data set1

Some of the data sets that were crawled down are shown in Figure 4.4 and Figure 4.5. Figure 4.4 shows the time data and the number of visitors to some of the tourist attractions and the maximum number of visitors that the tourist attractions can accommodate. Figure 4.5 shows the format of the weather data, including wind speed (corresponding to wind in the graph), rainfall (corresponding to rain), pm2.5 index (corresponding to pm25) and temperature. The frequency of data crawling is once a minute.

Considering that the system cannot currently include all tourist attractions in Shanghai, and that not all of them are already open, I have selected tourist attractions with different orientations on the map and different maximum visitor capacities as targets for the project study. Their position on the map is shown in Figure 4.6.

For convenience, I will refer to them as A, B, C, D, E and F. Their correspondence is shown in Table 4.1.

To ensure a sufficient amount of data for calculation and modelling, I collected 4767



B	EC	ED	EE	EF	EG	EH	EI
	wind	空气质量	pm25	能见度	rain	temperature	天气
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy
	0km/h	18	18	29km	0	25	Cloudy

Figure 4.5: Example of a partial data set2

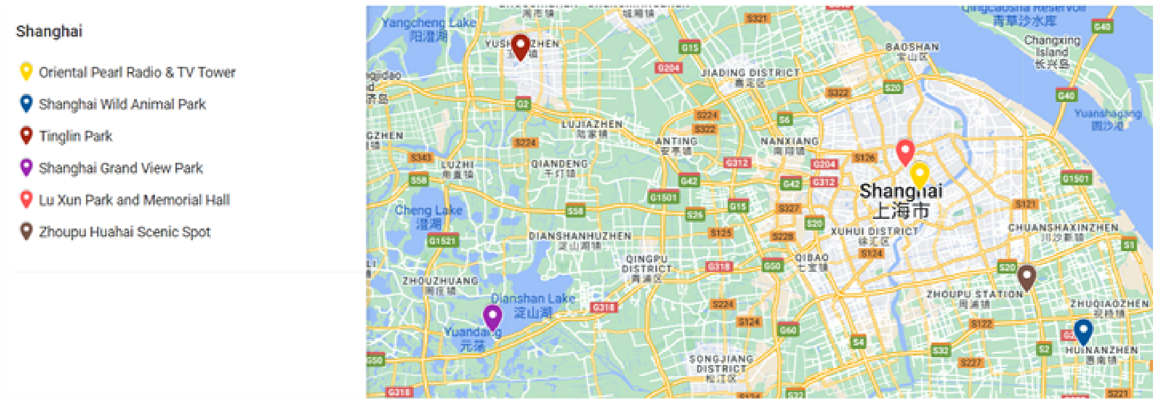


Figure 4.6: Distribution of selected tourist attractions on the map

Tourist Attractions	Designator	Maximum number of visitors
Oriental Pearl Radio and TV Tower	A	15000
Shanghai Wild Animal Park	B	96000
Tinglin Park	C	3500
Shanghai Grand View Park	D	5271
Lu Xun Park and Memerial Hall	E	32580
Zhoupu Huahai Scenic Spot	F	7500

Table 4.1: Caption that explains the table to the reader

rows of data, spanning four natural days. The invalid data were then removed and they were organised into the format of Figure 4.7.

	F	wind	pm25	rain	temperature
time					
2022-06-21 02:49:13.027422	0	1	15	0.0	25
2022-06-21 02:50:12.593289	0	1	15	0.0	25
2022-06-21 02:51:12.134381	0	1	15	0.0	25
2022-06-21 02:52:11.670767	0	1	15	0.0	25
2022-06-21 02:53:11.237718	0	1	15	0.0	25
...	..	...	...	...	...
2022-06-24 09:38:56.970700	0	1	25	1.6	25
2022-06-24 09:39:56.568608	0	1	25	1.6	25
2022-06-24 09:40:56.170003	0	1	25	1.6	25
2022-06-24 09:41:55.729960	0	1	25	1.6	25
2022-06-24 09:42:55.319916	0	1	25	1.6	25

Figure 4.7: Collated data sets

4.2.2 Modelling

The model for this project used an LSTM, but the data sets did not fit the model very well to begin with, so they needed some processing. The steps I took were:

1. Slice the test set and the training set

   First, the data set is sliced. The data set is sliced 1:2 into a training set and a test set. This resulted in 1559 rows of data for the training model and 3148 rows of data for the test.

2. Scaling the data

   As the data range is very large and they are not scaled in the same range, I use MinMaxScaler to scale the data in order to avoid prediction errors.

3. Splitting X and Y

   The number of tourists to be predicted is used as Y and the four weather data (feature values) are used as X. I then convert the X and Y lists into arrays and train them in LSTM model in an array format. I chose to use 30 bars of data to predict the 31st, after the first prediction it would automatically move to column 2 and take the next 30 values to predict the next target value. As it is necessary to predict the number of visitors per minute for the next 2 hours, i.e. 120 minutes, this process needs to be repeated 120 times.

4. Hyperparameter tuning

   I used girdsearchCV to perform some hyperparameter tuning to find the base model. This function will give the best parameters for the model.

5. Training the model and prediction

The processed data and optimal hyperparameters are brought into the model for training, and the model is used for prediction.

6. Restore the data and compare

   As the data used in training the model is scaled back, to get the true value, the predictions are expanded in reverse using the inverse_transform function. The predicted values are compared with the original values.

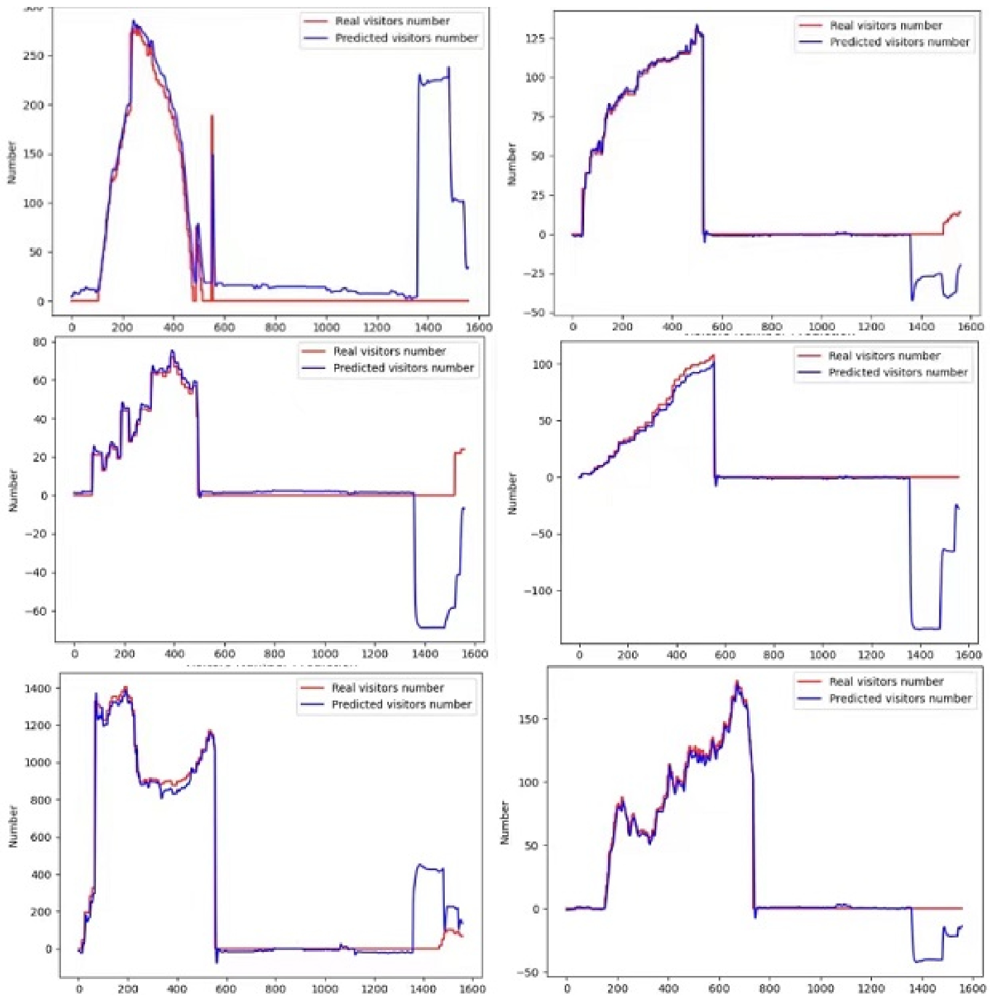


Figure 4.8: Comparison between model predictions and original values

Figure 4.8 below shows the comparison between the predicted and original values for the number of visitors to each tourist attraction (from top to bottom, left to right, corresponding to A, B, C, D, E and F). It can be seen that the fit between the predicted and original values is relatively good, so for each tourist attraction the model is available.

4.3 User interface

The project consists of 6 pages, corresponding to 6 tourist attractions. In the process of writing the pages, I generated QR codes for each page, where the QR code for A is shown in Figure 4.9, and the rest are similar to this, so I won't show them all. When the user scans the QR code, they will be able to access the page of the corresponding tourist attraction.

Figure 4.9: QR code for A

Each page includes current weather information, the number of visitors to that tourist attraction, a line graph of the predicted number of visitors, a table of recommended tourist attractions and a subscribe button.

As shown in Figure 4.10, to make it easier for visitors to use and to enable them to go to a destination page to view information about that destination without arriving at the destination, I have included destination selection buttons on each page.

Figure 4.10: Attractions switch button at the top of the page

Below, the user view of each function is shown separately.

4.3.1 Obtaining information on tourist attractions

As shown in Figure 4.11, the page shows the current attractions visited by visitors, the weather information at the time the page was visited, and the number of visitors.

Where the current visitor count status is marked with a different colour. If the current visitor count is less than 50% of the maximum visitor capacity of the attraction, the visitor

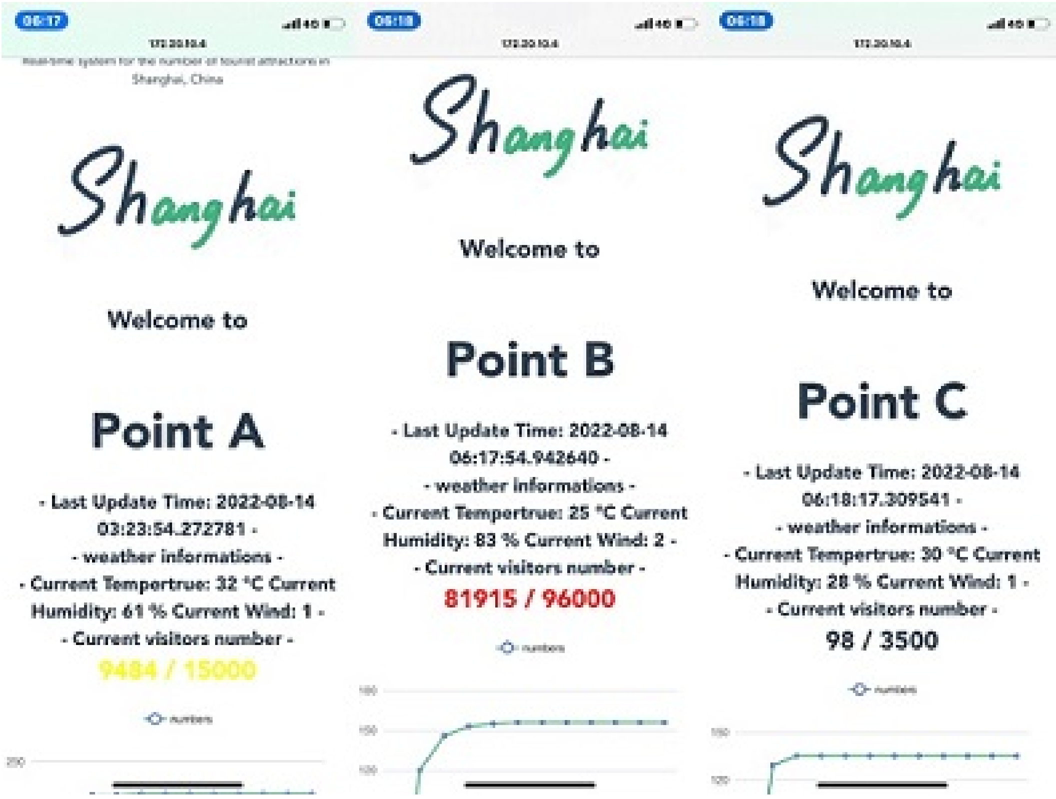


Figure 4.11: Weather information and number of visitors in different colours

count is marked in plain black. If the number of visitors is between 50% and 80%, it is marked in yellow, indicating that the attraction is currently crowded; if the number of visitors is above 80%, it is marked in red, indicating that the attraction is about to reach capacity and visitors are not recommended to continue.

4.3.2 Visitor number forecasting

The back-end server calls the model corresponding to the tourist attraction and makes a prediction of the number of visitors 120 minutes in the future from this point onwards, which is stored on the server. When the visitor visits the page, the server returns the predictions to the front-end as an array and renders them as a line graph on the page (as shown in Figure 4.12). This allows the visitor to consider whether or not to visit the attraction based on this line graph.

4.3.3 Tourist attraction recommendations

Tourist attraction recommendations are presented to visitors in the form of a list (Figure 4.13). Where p represents the availability of the recommended tourist attraction when

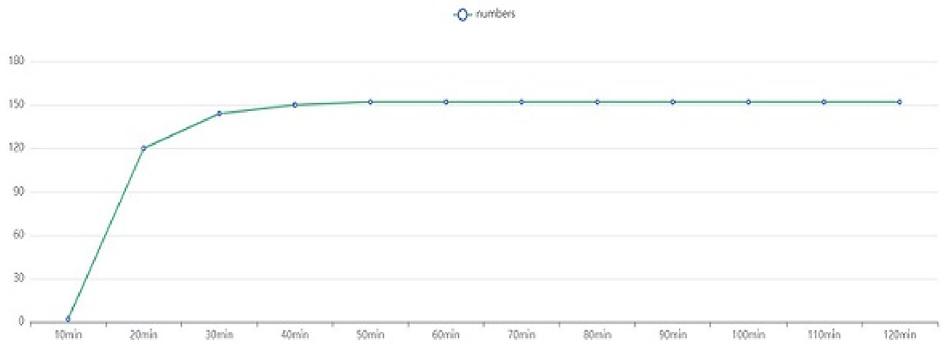


Figure 4.12: Visitor number forecast line chart

- Recommended Attractions -

No.	Name	p	Numbers
1	Piont B	99.71	16514
2	Piont E	99.13	20673
3	Piont A	96.98	1568

Figure 4.13: List of recommended tourist attractions

the visitor arrives at it from the current attraction and Numbers represents the current number of visitors to the recommended tourist attraction. If there are multiple attractions with less than 50% availability, the system will only recommend the three attractions with the highest availability in positive order to the visitor. If there are less than three, then a few will be recommended. This allows visitors to choose which tourist attraction to visit based on this list of recommendations.

Figure 4.14: Subscribe button

4.3.4 Attractions subscription

As shown in Figure 4.14, when a visitor clicks the subscribe button, the system will record the attraction. If the current attraction is predicted to have less than 50% of the total visitor capacity, a pop-up alert is sent to the user's page (Figure 4.15).

If the user clicks to unsubscribe Figure 4.16, they will no longer receive the alert.

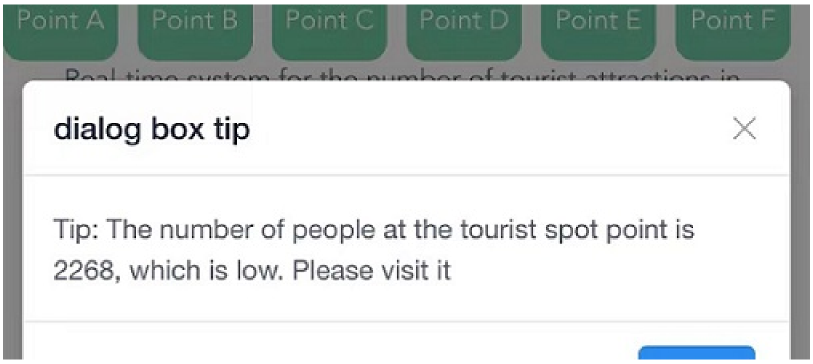


Figure 4.15: Weather information and number of visitors in different colours

Figure 4.16: Cancel subscribe button

4.4 Summary

This chapter documents the details of the system implementation in detail.

In section 4.1, the frameworks and technologies used for the system are indicated. The front-end framework is Vue.Js, the back-end framework is Django and the cloud server is Tencent CVM. In section 4.2, the process of the results of data collection and the process and results of building the LSTM model are documented. In section 4.3, each user interface of the web application is given. The main areas include weather information, visitor numbers, visitor predictions, attraction recommendations and a subscribe button.

The next chapter will present the system evaluation part of the project.

Chapter 5

Evaluation

After implementing the system functionality and confirming that the system is functioning correctly, the lstm model performance, and the performance of the web application, needs to be evaluated.

5.1 LSTM model evaluation

As this project uses lstm for forecasting, it can be considered as a regression model. Mean Squared Error (MSE), Root Mean Squard Error (RMSE) and Mean Absolute Error (MAE) are among the most common measures of continuous variables. In this paper, the performance of LSTM models is evaluated using these three models.

The formulae for these three models and their limitations are as follows:

Assume that the predicted value of the model is $\hat{y} = \{\hat{y}_1, \hat{y}_2, \hat{y}_3, ..., \hat{y}_n\}$

The true value of the data is $y = \{y_1, y_2, y_3, ..., y_n\}$

- MAE:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

  The range of MAE is $[0, +\infty)$, which is a perfect model when the predicted value exactly matches the true value, which is equal to 0; and the larger its error, the larger the value of MAE.

- MSE:

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

  The MSE reflects the true error of the result, which ranges from $[0, +/infty)$. That is, the MSE is equal to zero when the predicted value exactly matches the true value, and the larger the error, the larger the MSE.

- RMSE:

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

The range of RMSE is also $[0, +\infty)$, which is equal to 0 when the predicted value exactly matches the true value, and the larger the error between them the larger the value will be, and the larger the value of RMSE will be.

The MSE differs from the RMSE only in being sensitive to the magnitude, while the RMSE has the same magnitude as the MAE. However, comparing the RMSE and MAE, the RMSE will be somewhat larger than the MAE. The smaller the value of RMSE in the measure the more significant it is, as a smaller RMSE reflects the fact that the maximum error in the results is also smaller, which indicates a better fit of the model.

Another model mentioned in section 2.1.2: ARIMA, which has been used several times in several papers and also has good performance. I have chosen ARIMA to model and predict these same data and use it as a comparison with the LSTM model. Table 5.1 below shows the different performance metric values for ARIMA and LSTM, also for A. Due to the nature of the model, the values of the metrics obtained from each modelling exercise are different, so each value in the table is the average of 10 calculations performed for each.

Metrics	**ARIMA**	**LSTM**
MAE	23.9323	11.6837
MSE	1660.9593	700.5111
RMSE	40.7548	26.4671

Table 5.1: Different performance metric values for ARIMA and LSTM

As can be seen from the table, the ARIMA model is almost twice as good as the LSTM in all three metrics. It can therefore be determined that the LSTM performs better than the ARIMA model, which is also a time series model, in the data set of this project.

Therefore, the performance metrics of the LSTM (MAE=11.6837, MSE=700.5111, RMSE=26.4671) can be used as an evaluation criterion in this study for the reading public. And in future work, more layers can be considered to improve the LSTM and allow it to show better performance.

5.2 Web application evaluation

As the user interface for this project is based on a web application, it is necessary to evaluate the performance of the web application. The most time consuming steps of the system will be analysed first and then experimentally evaluated using mobile phones.

5.2.1 Analysis of time-consuming steps

Page response time is an important indicator of the performance of a web application. The two most time-consuming steps in this system are.

1. Line graph data calculation.

   As this system involves a prediction time span of 120 minutes, this means that to produce a predicted line graph of the number of visitors to a tourist attraction, 120 operations need to be performed backwards. This result is then reflected to the frontend for rendering.

2. Forecast for other tourist attractions.

   To derive the number of visitors to each of the remaining tourist attractions other than the current tourist attraction, separate calculations are required using the model. A tourist attraction that is close and not congested may take less time to calculate, and a tourist attraction that takes longer to reach will take longer to calculate. This is therefore an indeterminate length of time, which may or may not be very long.

Each time a page is visited or refreshed, two steps as described above need to be performed. The back-end calculations are therefore very heavy. To reduce the response time of the page and speed up the efficiency of displaying the charts, I set up timed tasks on the Django server to automatically perform model calculations every 15 minutes. This returns the results of the last calculation to the front-end page when the data request reaches the server. This improves the efficiency of the page display to a certain extent, but also makes the predictions differ from the true values with some latency.

5.2.2 Experiments and results

The tools used in this project to evaluate the performance of web applications are the iphone 11 and the Vivo X9. The iphone 11 is a relatively new phone, which was produced in September 2019. The Vivo X9, on the other hand, represents the older smartphone, which was produced in November 2016.

Frames per second experiments

Table 5.2 shows the hardware details of these phones, as well as the experimental results of the average frame rate expressed in frames per second (FPS).

Device	**SoC**	**CPU**	**RAM**	**ROM**	**FPS**
iphone 11	A13 Bionic SoC	6 cores 2.66GHz	4GB	128G	24
Vivo X9	Snapdragon 625	8 cores 2GHz	4GB	64G	21

Table 5.2: Iphone 11 and Vivo X9 performance details

The results of the iphone 11 test showed that the iphone 11 had a higher FPS than the Vivo X9 when it came to opening pages. however, no significant lag was felt during use on either the newer or older phones. Therefore, on some older phones, as long as the browser is working properly, pages will open without any problems, although perhaps not very quickly.

Time experiments

Due to the short development time and the limited resources of the phones, the system could not be tested on a large number of smartphones. Instead, this experiment recorded the time it took to open pages from different tourist attractions 50 times at random on each of the two phones. The results were thus used to roughly estimate how long it would take to load a page on most phones.

Table 5.3 shows the results for the iphone 11 and Vivo X9.

	iphone 11	**Vivo X9**
Shortest time (s)	2.36	2.11
Longest time (s)	3.19	2.37
Standard Deviation	0.1	0.1
Mean time (s)	2.51	2.28
95% Confidence Interval	$\pm$ 0.03	$\pm$ 0.02

Table 5.3: Time consumption statistics

As can be seen from the results, the difference between the two phones in terms of time taken to access and fully load the page is not significant and is within a reasonable range. The measures mentioned in the previous section had the effect of speeding up the loading of the pages.

In future work, if the opportunity arises, plans will be made to test how the system performs on more different devices and to use more precise experimental tools and instruments to obtain more accurate results. In addition, future research may evaluate more metrics on computational efficiency, such as CPU and GPU usage. A more in-depth evaluation could further reveal performance bottlenecks in a model or program task so that future research can be more specific in optimising the system.

5.3 Summary

This chapter contains the experiments and results of the evaluation of the system.

Section 5.1 is devoted to the evaluation of the LSTM model. The metrics chosen for the evaluation are MSE, RMSE and MAE for the regression model and the results of the LSTM model are compared with the results of the most basic time series model, the ARIMA model, and it is found that the LSTM model performs better. Section 5.2 evaluates the performance of the web application. The experiments were conducted using two phones, an iphone 11 and a Vivo X9, both new and old, for fps and page opening speed, both of which indicated that the application works fine.

The next chapter summarises the full dissertation and presents future work.

Chapter 6

Conclusions & Future Work

The first chapters describe the various parts of the system in terms of design and implementation respectively. Chapter 5 documents the experiments and results of the evaluation of the performance of the LSTM and the web application, and provides a brief critical analysis. In this chapter, however, the dissertation will summarise the evaluation results and issues in the design and development of the project, and draw research conclusions and suggest possible future work in this area.

6.1 Project contribution

The evaluation results validated that it was possible to put this tourist attraction recommendation system into use in Shanghai. By contributing greatly to this project and the dissertation, I have finalised the system. The following points are the contributions of this project.

- The exploration of the background and purpose of the project revealed the need for a tourist attraction recommendation application based on real-time visitor numbers for tourists travelling in Shanghai in the post-epidemic era.
- This project investigated and reviewed many previous studies on headcounting, real-time headcount prediction, headcount balancing methods and tourist attraction recommendation systems.
- This project designs and implements a web-based system to provide users with current visitor information for tourist attractions.
- This system applies LSTM model for real time visitor number prediction and designs and implements tourist attraction recommendation rules and attraction subscriptions.

- This system was ported to run continuously on CVM to enable visitors to scan QR codes to access pages at any time.
- This project evaluates the LSTM model with multiple metrics and tests the performance of its use on two mobile phones, the iphone 11 and the Vivo X9.

6.2 System limitations

Due to the short time frame of this project, there are still many issues that remain to be solved and areas for further improvement in the design and development process. The main limitations of this system are those stated below.

- I did not travel to Shanghai while working on this project as the conditions did not allow me to do so, so my hardware section only presents the theory and verifies the functionality of collecting data. The system was not field tested and performance experiments were not conducted in Shanghai. Therefore the dataset I used for modelling was publicly available data crawled from the web, and the data used for testing the functionality of the system was some reasonable random numbers generated by the back-end server.
- The data collection for this project was not perfect. Because Shanghai has just opened as a tourist attraction and has been in operation for a short period of time, none of the visitor numbers were regularly distributed in the interval from 0 to the maximum number of visitors accommodated during the project. This resulted in the real number of visitors being within 400, while the server simulations sometimes reached tens of thousands, so the accuracy of the LSTM was not evaluated when the number of visitors exceeded 400.
- Due to the wide distribution of several tourist attractions selected for this project, some of them are very far apart from each other. When calling the map API for time estimation, large values may be obtained. This means that if a tourist arrives at the tourist attraction in theory, it may have already exceeded the opening hours of the attraction.
- The way the system currently uses to reduce page load speed does not solve the problem at source, but rather visually speeds up the opening of the page by creating a delay. The root cause of this is that the model is slow and must be run multiple times. Therefore a more efficient system architecture and model structure are both elements worthy of continued research in subsequent work.

- In the performance testing phase, the results of the evaluation experiments can only be used as a reference due to the lack of more professional and accurate measurement tools and the lack of a sufficient number of different experimental mobiles. If a sufficient number of visitors are invited to volunteer for the system use evaluation experiment, this should also be done through Ethics Application, which is something I have not done yet.

6.3 Future work

Addressing the limitations mentioned in section 6.2 will be part of the future work. Another part of the future work will be to complement the unfinished parts of the system, such as performing system security considerations and design. These include four main areas, namely hardware, models and predictions, system performance and security.

In terms of hardware, if conditions permit, the short-term work should be to use a small closed environment for testing. Hardware testing is used to identify possible problems with the hardware and to link it to the software part. The long-term goal will be to set up and test in the field in several real tourist attractions.

For the model and predictions, in three months a large amount of data can continue to be collected and attempts can be made to increase the complexity and depth of the LSTM and explore better model structures to obtain more accurate predictions. In six months the LSTM can be replaced with other models, such as the gray model mentioned several times in the literature review section, to explore models that are more suitable for Shanghai's tourist attractions.

For the system itself, work in the short term should be directed towards optimising the system structure and thinking about how to increase system stability and responsiveness from an engineering perspective. The long-term plan is to increase the scalability of the system to include more tourist attractions in Shanghai, rather than limiting it to just the current six. Given more time, the system could be rolled out to any city in the world.

Finally, it is also very important to ensure that the system is secure. As the test experiments are gradually expanded and improved, and as they may be put into use in the future, future work should consider common cyber-attacks such as XSS, DDoS and DNS hijacking. And a system pressure resistance test should also be carried out to understand the limits of the system and to develop solutions.

Bibliography

Al-Deek, H. M. (2002). Use of vessel freight data to forecast heavy truck movements at seaports. *Transportation Research Record*, 1804(1):217–224.

Antoniou, C., Ben-Akiva, M., and Koutsopoulos, H. N. (2007). Nonlinear kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):661–670.

Badr, K., Iqbal, K., Mahboob, S., and Khan, M. Y. (2021). Processing fire detection, face recognition, and head counting for an iot-based application in emergency exit. In *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pages 1–6.

Barnich, O. and Droogenbroeck, M. (2011). Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20:1709 – 1724.

Chandra, S. R. and Al-Deek, H. (2009). Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, 13(2):53–72.

Chengcai, T., honglei, Z., Lei, Z., Yuanyuan, Y., and Ge, W. (2022). The impact of the new pneumonia epidemic on china's tourism industry and its response. *Ecotourism in China*, 12(1):169.

Cho, E., Myers, S. A., and Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090.

Coşkun, A., Kara, A., Parlaktuna, M., Ozkan, M., and Parlaktuna, O. (2015). People counting system by using kinect sensor. In *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, pages 1–7.

Daganzo, C. F. and Sheffi, Y. (1977). On stochastic models of traffic assignment. *Transportation science*, 11(3):253–274.

Dan, B.-K., Kim, Y.-S., Suryanto, Jung, J.-Y., and Ko, S.-J. (2012). Robust people counting system based on sensor fusion. *IEEE Transactions on Consumer Electronics*, 58(3):1013–1021.

Depatla, S. and Mostofi, Y. (2018). Passive crowd speed estimation and head counting using wifi. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9.

Fu, H., Ma, H., and Xiao, H. (2012). Real-time accurate crowd counting based on rgb-d information. In *2012 19th IEEE International Conference on Image Processing*, pages 2685–2688.

Fu, R., Zhang, Z., and Li, L. (2016a). Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328.

Fu, R., Zhang, Z., and Li, L. (2016b). Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE.

Gao, H., Tang, J., Hu, X., and Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100.

Hashimoto, K., Morinaka, K., Yoshiike, N., Kawaguchi, C., and Matsueda, S. (1997). People count system using multi-sensing application. In *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*, volume 2, pages 1291–1294 vol.2.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1):185–203.

Hua, J. and Faghri, A. (1994). Apphcations of artificial neural networks to intelligent vehicle-highway systems. *Transportation Research Record*, 1453:83.

Jahn, O., Möhring, R. H., Schulz, A. S., and Stier-Moses, N. E. (2005). System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations research*, 53(4):600–616.

Jia, T. and Junyi, L. (2016). Study on the intra-day time distribution pattern of domestic tourists in xi'an based on microblogging data. *human geography*, 31(3):151–160.

Jiang, N., Xie, C., and Waller, S. T. (2012). Path-constrained traffic assignment: model and algorithm. *Transportation Research Record*, 2283(1):25–33.

Jing, Y., Hu, H., Guo, S., Wang, X., and Chen, F. (2021). Short-term prediction of urban rail transit passenger flow in external passenger transport hub based on lstm-lgb-drs. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4611–4621.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45.

Karlaftis, M. G. and Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):387–399.

Kim, J.-W., Choi, K.-S., Choi, B.-D., Lee, J.-Y., and Ko, S.-J. (2003). Real-time system for counting the number of passing people using a single camera. In *Joint Pattern Recognition Symposium*, pages 466–473. Springer.

Kim, S.-Y., Cho, J.-H., Koschan, A., and Abidi, M. A. (2010). 3d video generation and service based on a tof depth sensor in mpeg-4 multimedia framework. *IEEE Transactions on Consumer Electronics*, 56(3):1730–1738.

Li, B., Zhang, J., Zhang, Z., and Xu, Y. (2014). A people counting method based on head detection and tracking. In *2014 International Conference on Smart Computing*, pages 136–141.

Li, R., Rai, L., Wang, H., and Liu, F. (2021). Forecasting of provincial tourist population based on grey neural network. In *2021 IEEE 4th International Conference on Electronic Information and Communication Technology (ICEICT)*, pages 790–793.

Lian, D., Chen, X., Li, J., Luo, W., and Gao, S. (2021). Locating and counting heads in crowds with a depth prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Lingbin, Y., Xia, Z., and Jin, W. (2009). Prediction of the number of international tourists in china based on gray model (1,1). In *2009 International Conference on Electronic Commerce and Business Intelligence*, pages 357–360.

Lingras, P., Sharma, S., and Zhong, M. (2002). Prediction of recreational travel using genetically designed regression and time-delay neural network models. *Transportation Research Record*, 1805(1):16–24.

Liu, J., He, Z., and Ma, S. (2018). An n -path logit-based stochastic user equilibrium model. *IEEE Access*, 6:20971–20986.

Liu, X., Tu, P. H., Rittscher, J., Perera, A., and Krahnstoever, N. (2005). Detecting and counting people in surveillance applications. In *IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, pages 306–311. IEEE.

Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197.

Migliorini, S., Carra, D., and Belussi, A. (2021). Distributing tourists among pois with an adaptive trip recommendation system. *IEEE Transactions on Emerging Topics in Computing*, 9(4):1765–1779.

Orrite-Uruñuela, C. and Vicente-Dueñas, D. (2018). Counting people by infrared depth sensors. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Park, D. and Rilett, L. R. (1999). Forecasting freeway link travel times with a multilayer feedforward neural network. *Computer-Aided Civil and Infrastructure Engineering*, 14(5):357–367.

Park, D., Rilett, L. R., and Han, G. (1999). Spectral basis neural networks for real-time travel time forecasting. *Journal of Transportation Engineering*, 125(6):515–523.

PENG, K., BAI, W., and WU, L. (2020). Passenger flow forecast of railway station based on improved lstm. In *2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, pages 166–170.

Segen, J. (1996). A camera-based system for tracking people in real time. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 63–67. IEEE.

Shahzad, A. R. and Jalal, A. (2021). A smart surveillance system for pedestrian tracking and counting using template matching. In *2021 International Conference on Robotics and Automation in Industry (ICRAI)*, pages 1–6.

Smith, B. L., Williams, B. M., and Oswald, R. K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4):303–321.

Song, S. and Zhou, Y. (2019). Analysis and prediction of passenger flow differences in shanghai 3a scenic spots and above based on arima model. *Statistics and Applications*, 8(3):537–552.

Sun, Y., Leng, B., and Guan, W. (2015). A novel wavelet-svm short-time passenger flow prediction in beijing subway system. *Neurocomputing*, 166:109–121.

Teixeira, T. and Savvides, A. (2008). Lightweight people counting and localizing for easily deployable indoors wsns. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):493–502.

Tian, Y. and Pan, L. (2015a). Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 153–158.

Tian, Y. and Pan, L. (2015b). Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, pages 153–158. IEEE.

Van Lint, J. (2006). Incremental and online learning through extended kalman filtering with constraint weights for freeway travel time prediction. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 1041–1046. IEEE.

Wang, X., Leckie, C., Chan, J., Lim, K. H., and Vaithianathan, T. (2016). Improving personalized trip recommendation by avoiding crowds. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 25–34.

Wang, Y. and Papageorgiou, M. (2005). Real-time freeway traffic state estimation based on extended kalman filter: a general approach. *Transportation Research Part B: Methodological*, 39(2):141–167.

Wardrop, J. G. (1952). Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 1(3):325–362.

Williams, B. M. and Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672.

Yam, K.-Y., Siu, W.-C., Law, N.-F., and Chan, C.-K. (2011). Effective bi-directional people flow counting for real time surveillance system. In *2011 IEEE International Conference on Consumer Electronics (ICCE)*, pages 863–864. IEEE.

Yang, Y., Altschuler, B., Liang, Z., and Li, X. R. (2021). Monitoring the global covid-19 impact on tourism: The covid19tourism index. *Annals of Tourism Research*, 90:103120.

Yin, H., Wong, S., Xu, J., and Wong, C. (2002). Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C: Emerging Technologies*, 10(2):85–98.

Yoon, H., Zheng, Y., Xie, X., and Woo, W. (2010). Smart itinerary recommendation based on user-generated gps trajectories. In *International conference on ubiquitous intelligence and computing*, pages 19–34. Springer.

Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270.

Yuan, J., Zhong, L., and Li, X. (2009). Research and development of grey neural networks. *Journal of Wuhan University of Technology*, 3.

Yuan, Q., Cong, G., Ma, Z., Sun, A., and Thalmann, N. M. (2013). Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372.

Zhang, J.-D. and Chow, C.-Y. (2015). Ticrec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations. *IEEE Transactions on Services Computing*, 9(4):633–646.

Zheng, Y. and Xie, X. (2011). Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):1–29.

Zin, T. T., Tin, P., Hama, H., and Toriu, T. (2011). Unattended object intelligent analyzer for consumer video surveillance. *IEEE Transactions on Consumer Electronics*, 57(2):549–557.