AI PROJECT REPORT
Bike Rental Management System and Traffic Congestion Prediction

Instructor Assoc.Prof. Hai Pham Van[1], Student Le Ky Anh[2], Student Le Anh Quan[2]
[1] School of Information Technology and Communication, Hanoi University of Science and Technology, Hai Ba Trung, Hanoi, Vietnam.
[2] Hanoi University of Science and Technology, Hai Ba Trung, Hanoi, Vietnam.

Abstract. The recent growth of metropolitan areas has increased demand for mobility while also raising serious concerns among authorities about traffic congestion. To answer these demands, many bike rental systems and traffic congestion prediction methods have been put forth. Despite that, traditional bike rental systems do not seem to be able to meet the increasing demand anymore. Therefore, in this paper, we employ software to automate the bike rental system and the temporal graph convolutional network (T-GCN) for traffic forecasting. The automatic bike rental system consists of three entities (User, Manager, and Staff), using HTML, CSS, NodeJS, and the database management system MongoDB to store and access data quickly. The T-GCN model is a combination of the graph convolutional network (GCN) and gated recurrent unit (GRU). The model is evaluated with the traffic dataset of a city in China.

Keywords: Bike System, Traffic Congestion, Traffic Prediction, LSTM, Machine Learning

1 Introduction

Traffic congestion has received a great deal of attention in recent years because it has a great impact on people's daily lives. Detailed congestion forecasts for each road segment in an urban area can help people plan their travel routes and support traffic control to reduce congestion. Therefore, it is important to design an accurate congestion prediction model at the granularity of road segments.

Benefits of the problem: Predicting future traffic conditions makes traffic coordination more convenient in avoiding road congestion. As well as providing information for everyone to find the right route for their move. From there, the move will take place more smoothly, with less cost.

Difficulties of the problem: Predicting traffic behavior is a complex problem. Traffic conditions are influenced by complex factors of time and space.

Spatial complexity: Changes in traffic volume are affected by the structure of the traffic network (e.g., the traffic volume of an intersection will be affected by the traffic volume of the connecting roads). in it).

Each component of the transport systems must function properly for effective transportation (e.g., public buses, public rapid rail, taxis, and private automobiles) and must be well connected. Because bicycles can be used to travel to a range of locations, a bicycle or bike-sharing system (BSS) can play an important part in achieving this goal. BSSs have many advantages over other forms of transportation, including accessibility, the reduction of air pollution, the promotion of good health, and cost-effectiveness[1], [2]. Currently, on the market, there are many solutions to solve the need for traveling on short distances. There are many solutions but not convenient for users. The current bike rental and loan mechanism is too complicated, and people need to intervene and many steps in the bike rental process. Through business analysis and the advantages and disadvantages of current methods, we realize that building a bike rental system is essential, both for users and for bike rental businesses. The system not only helps users choose the type of vehicle they need, but most importantly, the user can order and use the vehicle whenever the user wants. Besides, it also helps businesses to control the status and health of the business when the automatic bike rental solution aims to digitize this rental job, helping businesses manage the quantity as well as the number of customers. quality of service through the reports of each bus station. The required functions of the system are presented below:

Functions for the manager:

- Manage employees in the system
- Manage vehicles circulating in the system
- Manage current vehicle owners
- user management

Functions for staff/receptionists:

- Authenticate bike pick up/return user
- Manage existing vehicles at the parking lot
- Deposit money into user account

Functions for users:

- Book and rent a bike remotely
- Pick up and return the bike automatically
- Deposit money into your account
- Verify user information

From the functional requirements for the automatic bike rental system analyzed above, the system built should be fully functional to help candidates and managers operate most conveniently. Web-based applications will help managers easily access by personal computers or phones. For users, now everyone has a personal phone, so designing an app to personalize the bike rental experience for users is the most optimal.

The project to build a website to manage the automatic bike rental system is designed in two main parts: the front end is the part that directly interacts with the user of the application using HTML, CSS to display a combination of JavaScript to help oversee the application. To manage user operations, the backend uses NodeJS language to help process user requests according to the enrollment business process and send data back to the frontend. At the same time, the system uses the MongoDB database management system to store and access data quickly.

2 Background

Globally, there are more people using BSSs. Due to the COVID-19 outbreak, many users choose BSSs over congested public transportation. BSSs have numerous advantages over other forms of transportation, some of which are listed below.

- **Accessibility:** Public transportation services, such as buses and trains, are only available during specific hours and cannot go to many different areas. Contrarily, bicycles do not have set schedules and can be utilized to travel to a larger range of destinations.
- **Traffic**: Moving around in urban places is difficult because of the traffic. Cyclists don't have to worry about traffic, unlike those who utilize taxis or private vehicles to get almost anywhere.
- **Air pollution**: Environmental pollution is currently receiving a lot of attention, and many government programs take diverse environmental pollution issues into consideration. Bikes are a beneficial means of transportation regarding air pollution because they emit no gas.
- **Health and hobby**: Cycling is a healthy and fun form of transportation that is accessible to a wide range of people.
- **Price**: Comparatively cheaper than other modes of transportation is bike rental. For instance, employing a taxi service is highly costly.

Aside from the environmental benefits, improved city life quality, and a better experience in the usage of urban spaces, the deployment of bike sharing systems is projected to create some changes in mobility patterns. These services attract customers from other modes of transportation such as bus transit, walking, automobiles, and taxis. Furthermore, some authors argue that bike-sharing serves as both a competitor and a complement to existing modes of transportation because it may be used as an alternative to driving and journeys can be supplemented with other modes[3], [4].

BSSs can be divided into two sorts of systems: the free-floating system and the dedicated docking system. Users are only required to hire or return bicycles to specific places when using the dedicated docking system. The users of the free-floating system can return their bicycles wherever. Due to the excessive cost of station installation in the dedicated docking system, the BSS can only accommodate a certain number of stations. Although users can return bicycles to any point in the free-floating system, this can be inconvenient for pedestrians[2], [5].

3 Proposed Model

3.1 Traffic Congestion Prediction

3.1.1 Problem Statement

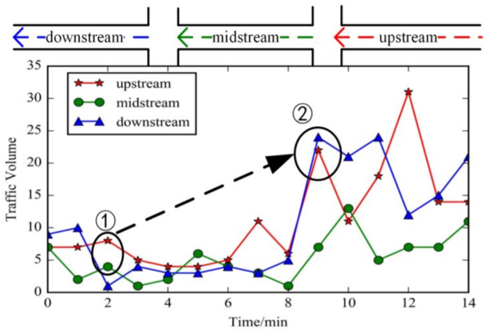
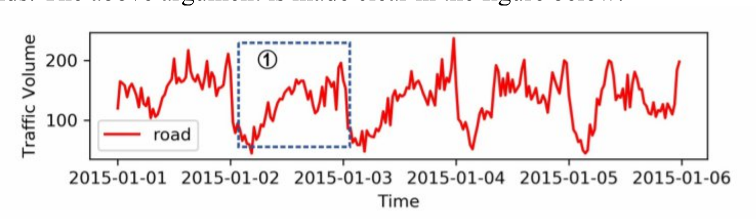


Figure 1: Spatial relation

As shown above, suppose there are three intersections: upstream node, midstream node and downstream node. The traffic of the upstream node affects the traffic of the downstream node through the intermediate node, and the traffic condition of the downstream node will negatively affect the traffic of the upstream node. As shown in Figure 1, we can clearly see the large influence of nodes that are close to each other.
Complexity: Vehicle traffic changes continuously over time, reflecting periodicity and trends. The above argument is made clear in the figure below:



(a)

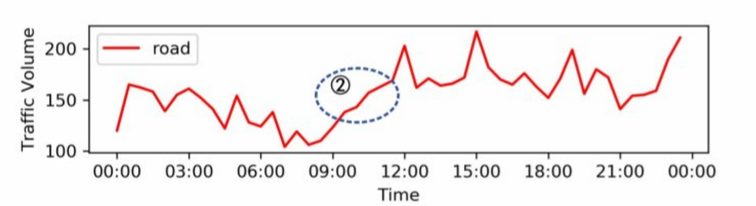


Figure 2

The first graph shows the change in traffic of route 1 for a week, and the second graph shows the change in a day. It can be clearly seen in the first graph that it is cyclical in nature and in intraday variations it is difficult to predict.
Modelization of the vehicle prediction problem is divided into the following three types of models:

Macroscopic Model	Mesoscopic Model	Microscopic Model
+ Macro model is a mathematical model that rep-	+ Mesos model will use groups of vehicles as a	+ Micro model simulates each vehicle.

resents the aggregate relationship of vehicle density, average speed of vehicles. The Macro model originates from the assumption that the behavior in traffic is similar to that in the fluid.	point for calculating vehicle traffic and velocity of that point. + Some Mesos models use cells (fixed length road segments) to group vehicles. + Movement in groups of vehicles based on the relationship about Macro	+ The variables of the model represent the position and speed of each vehicle Because it requires large computation time. So it limits the size of the transport network

Table 1: Modeling the problem

There are two main approaches to the traffic situation prediction problem. One is model-base driven. The second is data driven. The advantages and disadvantages of each method are shown in the table below:

	Model Base	Data Driven
Concept	1) Expert gives predictive model. Modeling with mathematical functions. 2) There are certain assumptions about the operation of the system 3) After having certain hypothesis about the model. The parameters will be fine-tuned based on the data	1) There is no need to hypothesize how the system works. Rules are extracted from the data.
Advantages	Fast execution. Because the model was released early Doesn't require too much computing resources	Because there is no need for a model, the properties of the problem are expressed in a more flexible way. Does not require too strict assumptions about the system
Opposite	Need proposed model before mathematical model. Sometimes the model proposed by the expert first does not match the properties of the problem. Then it will give bad results	Since the model is based on data, the quality of the model is affected by the quality of the data. Sometimes the data doesn't have a lot of noise. Requires a large amount of computational cost and computing resources.

Table 2: The two main approaches

The time division of the problem is divided into the following forms:
• Short-Term: Uses past traffic data to predict the next 15 minutes of future traffic.
• Medium-Term: Use data 2 hours ago to predict 2 hours later.
• Long-Term: Use the previous day's traffic data to predict the next day's traffic.

With our problem, we will be using the macroscopic model with the data-driven approach for short-term prediction.

3.1.2 Describing Data, Events and Knowledge

Description of Data

Unlike image, audio, and text data with a clear grid structure, the problem's data is more complex. As a result, the data representation needs to satisfy two properties:

- First: spatially, i.e., the geographical position of each node in space. Adjacent nodes (line segments) must be shown.
- Second: temporally, i.e., the change in traffic situations over time.

To satisfy the above two properties, the data will be presented in the form of a graph. The problem is only concerned with one property of the traffic system, namely, the average vehicle speed on each road segment. Because from the average vehicle speed on each road segment, we can deduce what condition that road segment is in. For example, if the average speed is less than 20 km/h, that section of the road is in a traffic jam. From 20 km/h to 30 km/h, it is in a traffic jam but can still move normally. As for the speed above 30 km/h, it can be considered open.

Spatial data: The transport system is represented by a directed graph G = {V, E}. We consider the intersection to be a node on the graph (as illustrated in figure 1, an upstream and a midstream are both nodes on graph G). The number of system intersections is then $V = \{v_1, \dots, v_M\}$. The edge (u,v) represents the intersection u adjacent to v (in other words, the road segment u can go to the road segment v), but the opposite direction is not necessarily true. We represent spatial relationships as adjacency lists.

Temporal data: represented by $X \in IR^{NxP}$representing the average speed of vehicles on the routes (nodes) over some time in the past. N is the number of traffic nodes as above. P is the number of consecutive periods in the past used. Then $X[v, t]$ represents the traffic volume of node v at time t in the past.

Purpose of the problem: With graphs G and X, we want to predict $Y \in IR^{NxL}$ where N is described as above and L is the number of 5 minutes in a row that we want to predict (in this case, 3). Alternatively, we want to find a mapping: $f(X, G) = Y' \in IR^{NxL}$ such that the two fathers of Y' *and* Y are as similar as possible.

Events and Knowledge

Events: To represent events, we need NxP variables representing the traffic of N nodes in the network over P consecutive periods in the past and NxL variables representing the traffic flow of N nodes over L consecutive periods in the future.

Knowledge: To predict traffic conditions, the essence is to find patterns hidden within the data. Knowledge is extracted from past data. The data here is the average velocity of each intersection, measured and stored.

From the average velocity data, we apply the following rules to deduce the traffic condition of a node (a route):

- If the average speed is less than 20 km/h, the road is considered blocked.
- If the average speed is between 20 km/h and 30 km/h, the road is in a traffic jam but can still move.
- If the average speed is more than 30 km/h, the road is clear.

Knowledge Demonstration

One of the most common ways to represent data is to represent relationships as a semantic network. Knowledge is represented in the form of a semantic network, shown in the figure below:

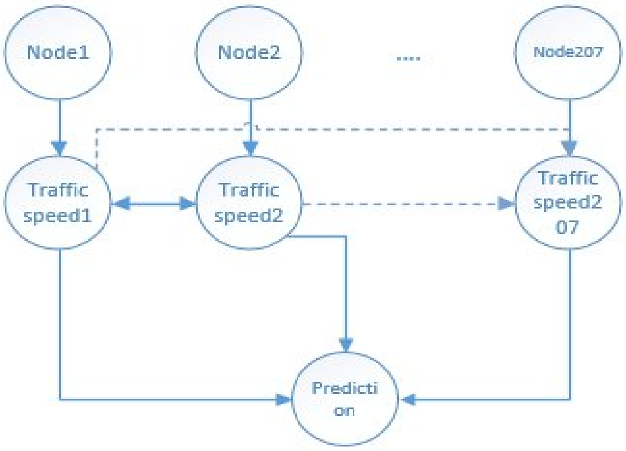


Figure 3: Semantic network representing the traffic system

The traffic system includes 207 traffic nodes. Each traffic node has a characteristic, which is the average speed of vehicles in 5 minutes. The average speed of vehicles on each node will be affected by the average speed of vehicles on other nodes. From the average speed of the past vehicles on the data nodes, the future average speed of 207 traffic nodes will be predicted.

3.2 Bike System

3.2.1 Business Process

Current information systems often have business processes ranging from simple to complex, which serve as a guideline for the entire system. The topic we built has an important professional process, which is enrollment management. Here are the specifics of this process:

Automated bike rental process

Process purpose	- The manager can control the current employees, vehicles and stations in the system - Employees could interact with user-agent accounts and events - Users recharge and rent bikes automatically
Agent	- User - Parking lot staff - Administrators
Process input	- User's bike rental request - Information between the current bus stations is in the system
Flow of events in the process	- User makes a deposit to the account - Users make bike rental - Show a list of parking stations - The user chooses the parking lot - Show vehicle list - User chooses a bike - The system authenticates the user who has successfully booked a bike - User takes the bike - Garage staff confirmed - User returns bike
Process output	- Users can book a bike - The administrator knows the statistics of bike rental orders
Sub event stream	Not available
Special Process	Not available

The automatic bike rental process consists of two main stages: the user makes a reservation for the bike and the user picks up/drops the bike. Details of the flow of activities in the process are presented in detail in the following diagrams (Activity_Pick up and return chart, Activity_Recharge chart, Activity_Rental chart).

3.2.2 Function Overview

Overview UseCase Diagram

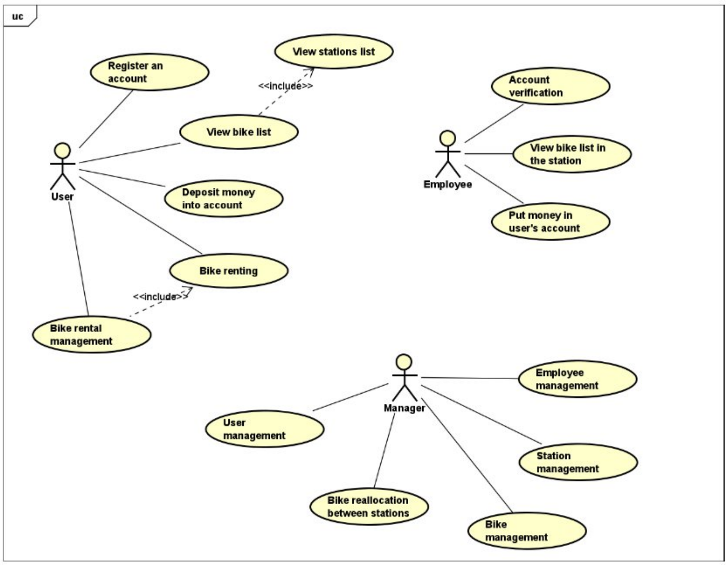


Figure 4.1: Overview UseCase Diagram

Station management UseCase decomposition diagram

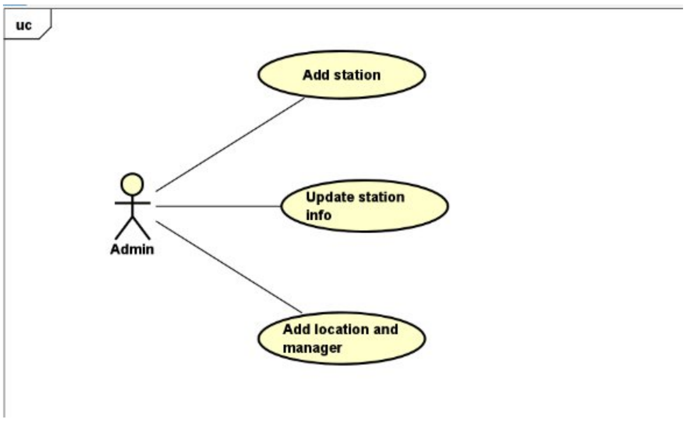


Figure 4.2: Station management UseCase diagram

The vehicle management use-case diagram depicts the parking management function of the manager. When the manager logs in to the system, he can view the information of the stations as well as make corrections and deletes of the existing parking lots. In addition, the manager can attach an employee to a bike station corresponding to the employee working at that station.

Employee management UseCase decomposition diagram

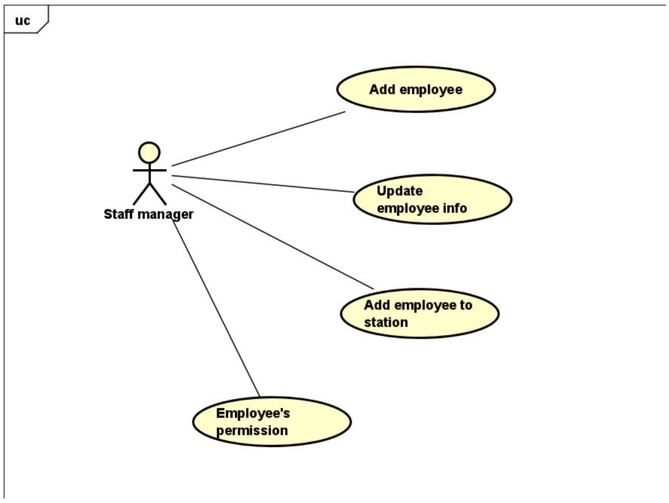


Figure 4.3: Employee management UseCase decomposition diagram

Employee management UseCase diagram depicts the employee management function of a manager. Managers can register accounts for employees to access the system using the UseCase of adding employees. In addition, the manager can decentralize the staff to perform recharge for users. Finally, the manager can attach an employee to any station so that the employee can access the system and manage the corresponding station.

Rental management UseCase decomposition diagram

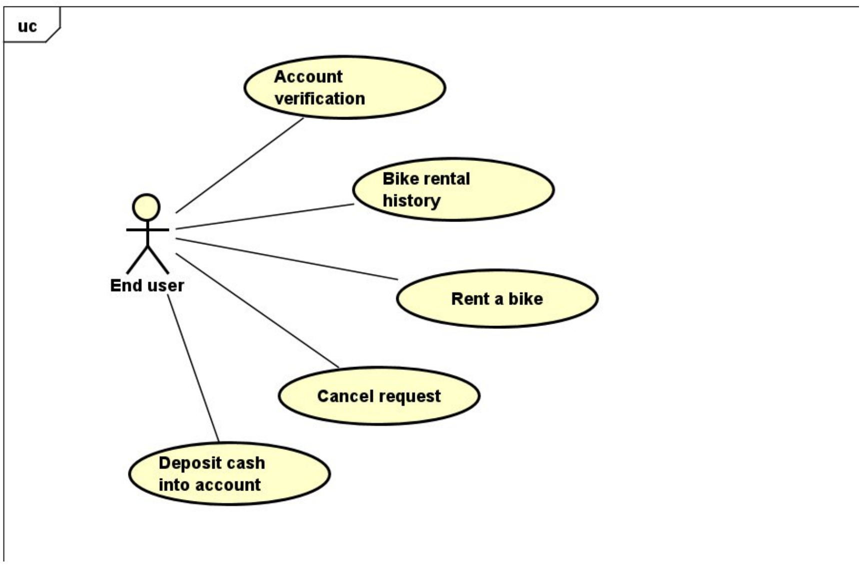


Figure 4.4: Rental management UseCase diagram

The booking management UseCase diagram describes the user's booking function. Users register for an account before they want to rent a bike, which happens automatically. After that, the user account will be unauthenticated, and the user must go to the nearest bus stop to meet the staff authorized by the manager to verify the account and top up the account. After authentication, users can perform car rentals and cancel operations at will. In addition, users are also provided with a car rental history function to help them control their accounts.

3.2.3 Functional specification

In this section, we will present the specification of the UseCase in the table below:

UseCase code	UseCase name
UC1	Deposit money into your account
UC2	Add station
UC3	Add staff
UC4	Bike rental
UC5	Bike return

Specification of UseCase to deposit money into your account

All users who want to perform the bike rental function must have a certain amount of money in their account if they want to use the service. Therefore, depositing money into the account is a necessary UseCase

UseCase code	UC1
UseCase name	Deposit money into your account
Actor's name	User
Describe	The use case describes a user who wants to make a deposit to an account
Pre-conditions	User has activated the account
Main event stream	1. The user moves to the nearest bus stop 2. User requests to staff to top up the account 3. Staff checking user's account system 4. Staff recharge user account
Sub event stream	Not available
Post-condition	User confirms successful deposit
Exception	3.1. User has not activated the account 3.2. The staff activates the user's account and continues to step 4

Frequency of use	Normal
Special requirements	Not available
Note	Not available

Specification of UseCase to add bike station

When the manager wants to deploy a certain station, the first condition is to add the new station to the system. Details of the UseCase are presented below

UseCase code	UC2
UseCase name	Add a station
Actor's name	Manager
Describe	The use case describes a manager who adds a new station and system
Pre-conditions	The administrator has logged into the system
Main event stream	1. The manager enters information about the parking lot 2. The manager chooses the location of the parking lot 3. The manager chooses the employees of the bike station 4. Confirm with the system
Sub event stream	Not available
Post-condition	Both the parking lot and the staff contain each other's information
Exception	2.1. Parking lot location does not exist 2.1.1. The manager added a new position 3.1. Administrator does not exist in the system UseCase stop
Frequency of use	Normal
Special requirements	Not available
Note	Not available

Specification of UseCase to add employees

The manager wants to add an employee to the system, use the add employee UseCase to do more and set up an account for the new employee

UseCase code	UC3
UseCase name	Add employees
Actor's name	Manager
Describe	The manager wants to add a new employee to the system
Pre-conditions	The administrator has logged into the system
Main event stream	1. The manager clicks the add employee button

	2. The manager enters the relevant information 3. Manager adds password for employees 4. New employee information authentication system
Sub event stream	Not available
Post-condition	New user existing account in the system
Exception	2.1. The relevant information already exists in the system 2.1.1. The manager fills in new information 3.1. Password does not meet system requirements 3.1.1 Selecting a new password
Frequency of use	Normal
Special requirements	Not available
Note	Not available

Specification of UseCase to rent bikes

Users want to make a bike reservation so they can pick up when needed, this is the UseCase that users will use the most

UseCase code	UC4
UseCase name	Bike rental
Actor's name	User
Describe	The manager wants to add a new employee to the system
Pre-conditions	User has logged into the system
Main event stream	1. User clicks and rents a bike button 2. User money checking system 3. User selects the parking lot 4. Users choose the type of bike they want to book 5. The system displays the user's request 6. User confirms successful booking
Sub event stream	Not available
Post-condition	User bike set state transition
Exception	2.1. User does not have enough funds in the account UseCase ends

Frequency of use	Normal
Special requirements	Not available
Note	Not available

Specification of UseCase to return bikes

After using the vehicle, the user returns the vehicle at an arbitrary station under the management of the system.

UseCase code	UC5
UseCase name	Bikes Return
Actor's name	User | Staff
Describe	User returns the vehicle and is authenticated by the employee
Pre-conditions	Employees log in to the system
Main event stream	1. The user arrives at the parking lot to pay 2. Scan the QR code for the system to identify the bus stop 3. The system displays a notice to return the vehicle to the corresponding station attendant 4. The staff authenticates the user to return the bike 5. User invoice update system 6. Staff reconfirms with users
Sub event stream	Not available
Post-condition	User bike set state transition
Exception	Not available
Frequency of use	Normal
Special requirements	Not available
Note	Not available

3.2.4 Non-Functional Requirements

Technical Requirements

The system needs to meet the technical requirements:

- The server responds fast enough to not create a bad experience for users
- The database must be secure

Convenience

- Users can use the system anytime, anywhere
- Users can easily get the bike quickly when they arrive at the bus station

Reliability

- User information is stored on the server and encrypted
- User accounts must be authenticated to be able to interact with the system
- Only authorized personnel can interact with users and the system

Binding Requirements

- Requires network connection
- Requires use of Chrome browser and phones with an updated version in the last 4 years

4 System Design

4.1 Traffic Congestion

4.1.1 Architectural model of the system

The architecture of the system is described in detail in the figure below:

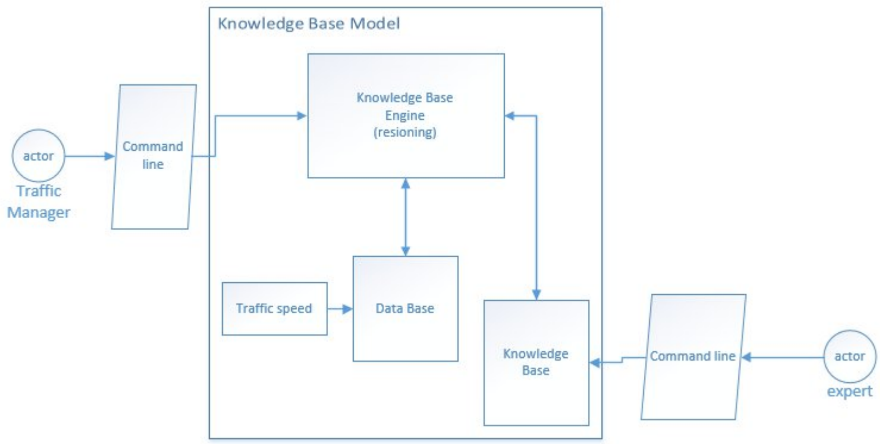


Figure 5: The system architecture

Traffic managers use the command line to communicate with the system. The Knowledge Base Engine uses a deep learning network architecture, described in detail in the algorithm section. A database is used to store information about the speed of vehicles in traffic nodes. The knowledge base is the set of parameters of the model after it has been trained with the data. This set was created by machine learning engineers.

4.1.2 Algorithm

Overview

The mathematical representation to calculate the traffic volume of the traffic system in the following time interval T can be represented by the following mathematical function:

$$\left[X_{t+1}, \dots, X_{t+T}\right] = f(G;(X_{t-n}, \dots, X_{t-1}, X_t))$$

To ensure that the complex nature of the data can be represented, the method used here is deep learning. Now the function f is used as a neural network.
Given n numbers of past periods and the number of future periods we want to predict, $X_i \in IR^N$ is an N-dimensional vector, each representing the traffic of N nodes in the traffic network at time t. G is the graph showing the spatial relationship of the transport system. The architecture of the neural network is shown below:

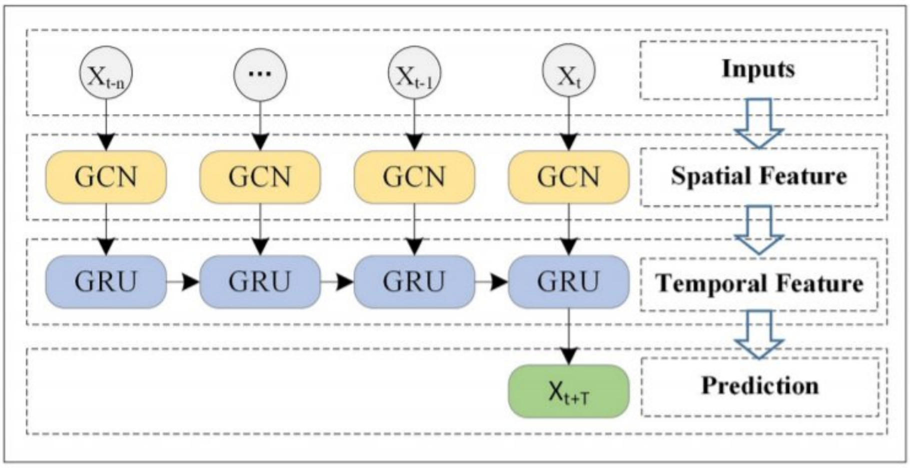


Figure 6: Overview of the model's architecture

The model uses two types of layers: Graph Convolution Layer for spatial feature extraction and Gate Recurrent Layer for temporal feature extraction. Details about these two types of layers will be clarified below.

Spatial model

Graph convolutional neural networks are used to solve complex spatial problems in transport systems. With the traditional Convolution operator, we can obtain local or global features. Though it can only be used in Euclidean space, such as in images or grid-structured data types. The traffic system network is a type of graph data, so it is not possible to apply CNN directly to this data type.

Graph convolution builds a filter to perform feature extraction on the nodes of the graph. By implementing it, we obtain spatial features between nodes.

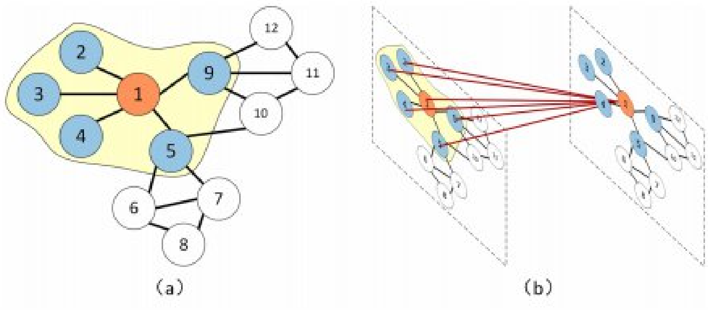


Figure 7: Graph convolution neural network

Graph convolution's operator is represented by the following mapping:

$$g(X, A) = sigmoid(A' \; relu(A' X W_0) W_1)$$

Where X is the spatial data matrix described in detail at the data description step. $A' = A + I$. A is the adjacency matrix representing the association relationship between nodes, and I is the unit matrix.

We use two activation functions here, $sigmoid$ and $relu$.

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

$$relu(x) = \begin{cases} 0 \; if \; x < 0 \\ x \; if \; x \geq 0 \end{cases}$$

The parameters to learn here are two matrices, W_0 and W_1.

Temporal model

To solve the temporal problem of series data, it is necessary to represent the change in traffic volume over time. Recurrent neural networks are the most widely used architecture in this context. However, this architecture has a major drawback in that when the prediction is too long, it will cause decent gradient suppression.

To overcome this drawback, the LSTM architecture is used.

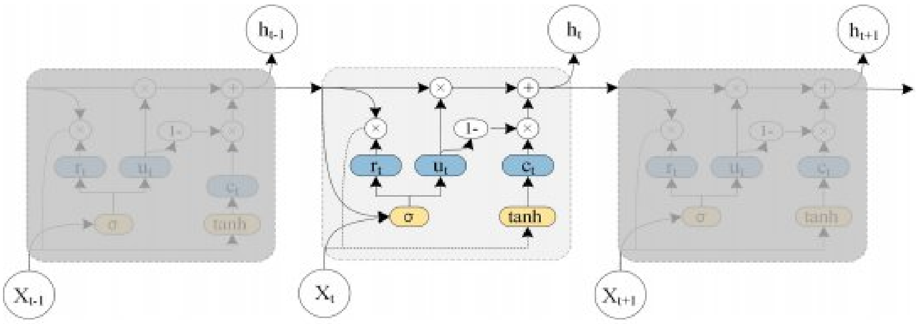


Figure 8: LSTM Architecture

h_{t-1} is the hidden state at time $t-1$. X_t is the output of the convolution graph at time t. u_t, r_t are the updated and deleted ports at time t. Parameters W, b are the parameters learned during training.

$$u_t = sigmoid(W_u\left[f\left(A, X_t\right), h_{t-1}\right] + b_u)$$
$$r_t = sigmoid(W_r\left[f\left(A, X_t\right), h_{t-1}\right] + b_t)$$
$$c_t = \tanh\left(W_c\left[f\left(A, X_t\right), r_t * h_{t-1}\right] + b_c\right)$$
$$h_t = u_t * h_{t-1} + \left(1 - u_t\right) * c_t$$

The tanh function is described as follows:

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

Due to the cyclic nature of the data, the architecture used is BiLSTM. BiLSTM will use two LSTMs. A forward LSTM represents the forward inference's relationship over time. A backward LSTM represents time-reversed inference. These two LSTMs' combined output will be the outcome.

Detailed architecture

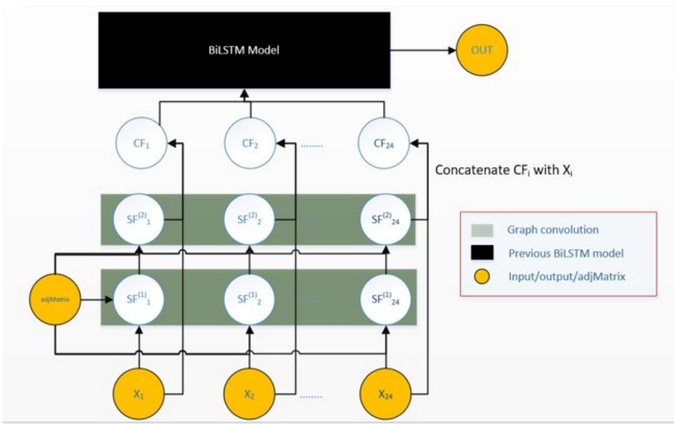


Figure 9a: Neural network architecture

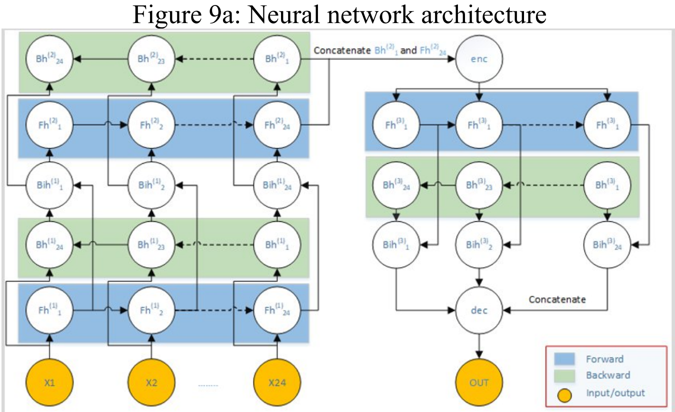


Figure 9b: LSTM architecture

The input is t time steps with each time step corresponding to a matrix $X_i \epsilon R^{N\times 1}$ where N is the number of intersections and 1 is the traffic speed features. $SF_i^{(k)}$ is the spatial information extracted at time step i through convolution layer k. $SF_i^{(k)} \epsilon R^{N\,x\,5}$. 5 is the number of hidden units used to represent information about neighbors. The BiLSTM model is built by stacking two consecutive BiLSTMs. Each LSTM in an LSTM has 64 hidden states.

After going through two BiLSTMs in the last step, we get a 128-dimensional vector to represent past time steps. The output is decoded using a BiLSTM layer with a dimension of 64

Loss Function

During training, the goal is to minimize the error between the actual value and the predicted value. Y_t *and* $Y\prime_t$ are used to describe the actual and predicted traffic values. The goal is to have the predicted value match the actual value as closely as possible.
In addition, due to the complex network architecture, the model is easy to overfit. To solve this problem, the technique used is L2 regularization.
The loss function is represented as follows:

$$loss = \frac{1}{2}||Y_t - Y'_t||^2 + lamda * ||w||_2^2$$

Optimal Algorithm

To minimize the loss function, the algorithm used is schochastic gradient descent. The algorithm is described below:

Pseudocode

```
For epoch = 0 to number of epochs - 1:
```

```
        For iter = 0 to number of iterations per epoch - 1:
            batch_x, batch_y = get_batch()
            l = loss(f(batch_x, A), batch_y)
              For i = 0 to number of parameters - 1:
                w[i] = w[i] - lr * ∂l/∂w[i];
            End For
        End For
    End For
```

The optimization algorithm is based on the first-order optimization condition. It is not guaranteed that the optimal value found is the global optimal point.

Measurements

1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_t - Y'_t)^2}$$

2) Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|(Y_t - Y'_t)|$$

3) Coefficient of Determination(R2):

$$R^2 = 1 - \frac{\sum_i (Y_t - Y'_t)^2}{\sum_i (Y_t - Y_{mean})^2}$$

4) Explained Variance Score (Var):

$$var = 1 - \frac{Var\{Y - Y'\}}{Var\{Y\}}$$

4.2 Bike System

4.2.1 Architectural Design

Choosing software architecture

The system is based on the MVC model, a design architecture that is commonly used in many software today. This model structure the source code into 3 main parts according to the tasks: processing, storage and display. The MVC model will support software development in parallel parts without affecting each other, ensuring the quality and progress of software development.

Components in MVC include:

- Controller: where to receive user requests through the view. Here, the thread will process complex logic, and at the same time get data and return it to the user. This is an important connection between the server and the client. For example, when a user submits a login request, the controller will get that login data, call the authentication model, and get the user information.
- Model: is where the business interacts with the data or the database management system. Model includes classes and functions that handle many operations such as database connection, input data validation, database query, add - delete - edit data. . .
- View: is the part that displays the user interface. View provides visual components such as images, information, data forms, buttons to send requests, ... View is structured mainly from HTML files.

When there is a request from the client side to the server, the controller will receive and call the corresponding model class to authenticate and get the data. After processing is complete, all results are returned by the controller to the client (view). In the view, the data will be used to generate HTML code, and the browser will use the HTML code to convert the information display interface.

Overall design

Here is the overview package chart in the project:

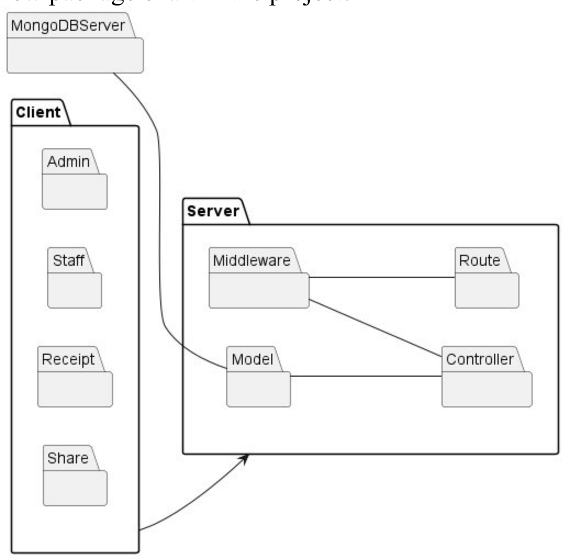


Figure 10: Overview package chart

The application packages have the following functions:

- On the client side:
  - The Admin package contains classes related to the admin interface

  - The Staff package contains files related to the employee's interface
  - The Receipt package contains files related to the reception's interface
  - The Share package contains commonly used interface files
- On the server side:
  - The Route package contains endpoints and configs for calling the respective controllers
  - The Controller package implements control and uses Models
  - The Middleware package is located between the Controller and the Route used to link between the two packages
  - The MModel package contains the model class that connects to the database
  - The MongoDBServer package is a symbolic package for the 3rd party that also provides the Database service

4.2.1 Detailed design

GUI design

The built application has actors such as manager, two levels of employees and users. Each agent has a separate function and a separate interface for each of those functions. But the layout of the application's interfaces is designed to ensure the principle of simplicity, creating an easy-to-see feeling.

Screen Info	Design
Screen resolution	Support for screen resolution 1080P and above
Supported screen size	15 inches or more
Color	Full RGB color support
Display notifications from the system	- Error message - Successful message - Warning message - Information Notice
Display language	- Vietnamese

Following are some main interface designs that illustrate the main functions of the system

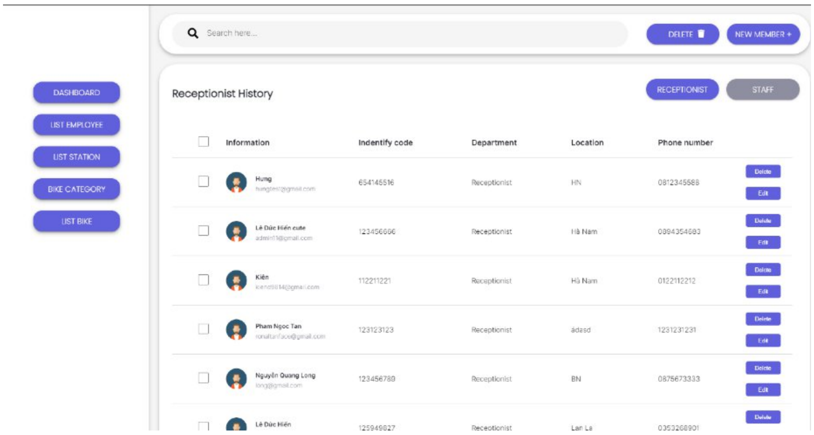


Figure 11.1: Employee management interface design

Figure 11.1 depicts the design of employee management. The interface includes navigation and search bars, a list of existing employees. Users when selecting an employee to view information, will display an interface to view detailed information about that employee and make relevant adjustments.

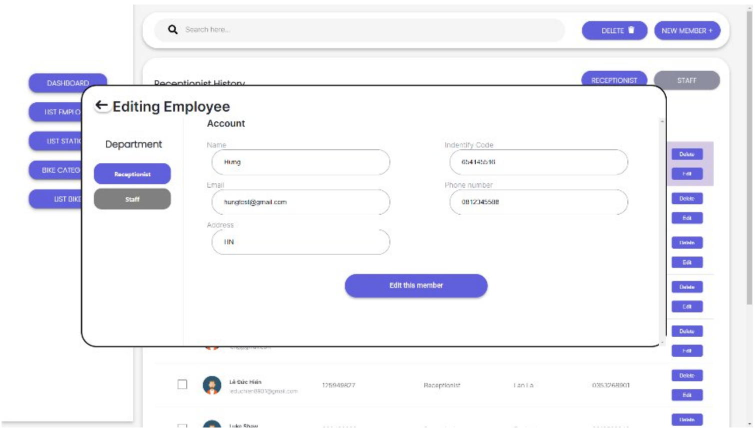


Figure 11.2: Employee information editing screen

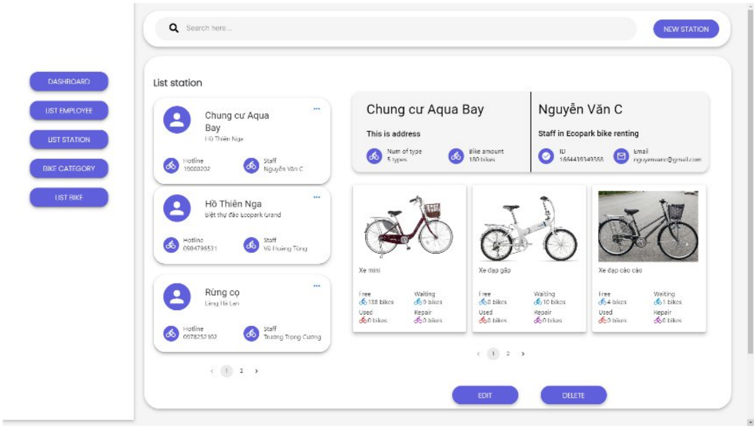


Figure 11.3: Management screen of parking lots

Figure 11.3 depicts the screen to manage the parking lots, the screen will display a navigation bar and a search bar for the manager to perform the parking search functions. When clicked on any bike, detailed information about that bus will be displayed right next to it, making it easy for the manager to track and make edits related to the bike.

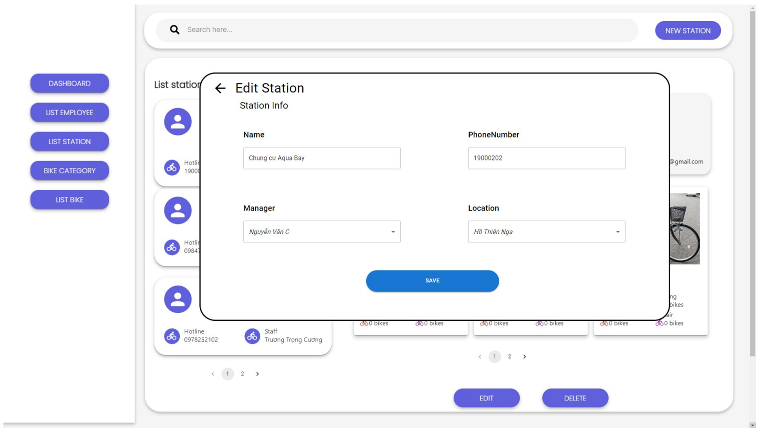


Figure 11.4: Station edits screen

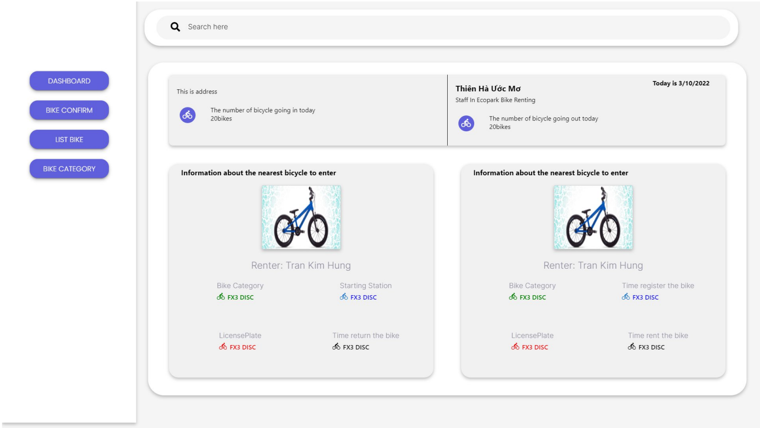


Figure 11.5: Employee vehicle access management screen

Figure 11.5 describes the employee's interface when the user makes a bike rental or returns a bike. Staff will see information related to the vehicle that the user is using and confirm the requirements related to the user's vehicle.

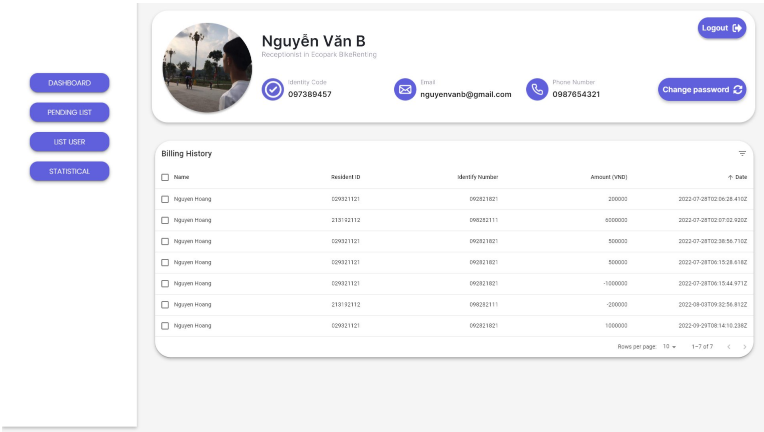


Figure 11.6: Main Screen of Receptionist

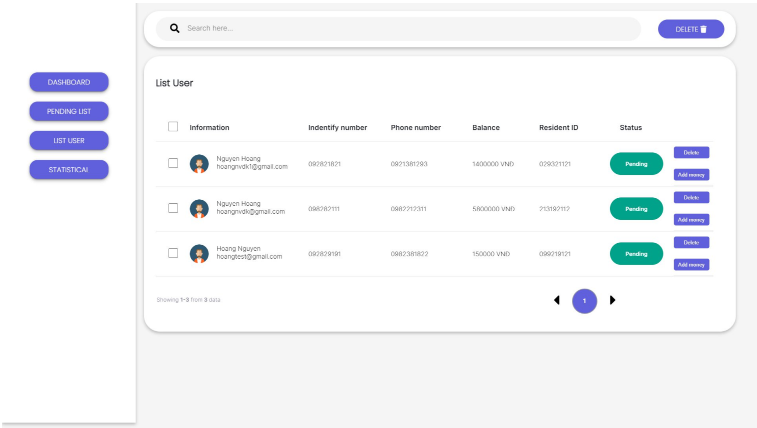


Figure 11.7 Employee user management screen

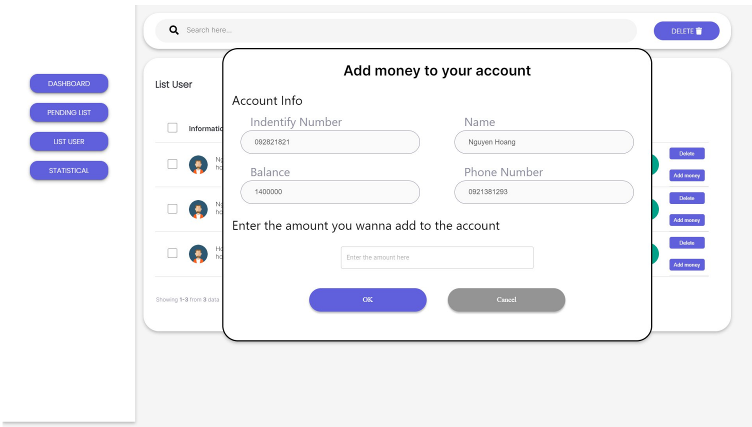


Figure 11.8 User account deposit screen

Database design

- User account

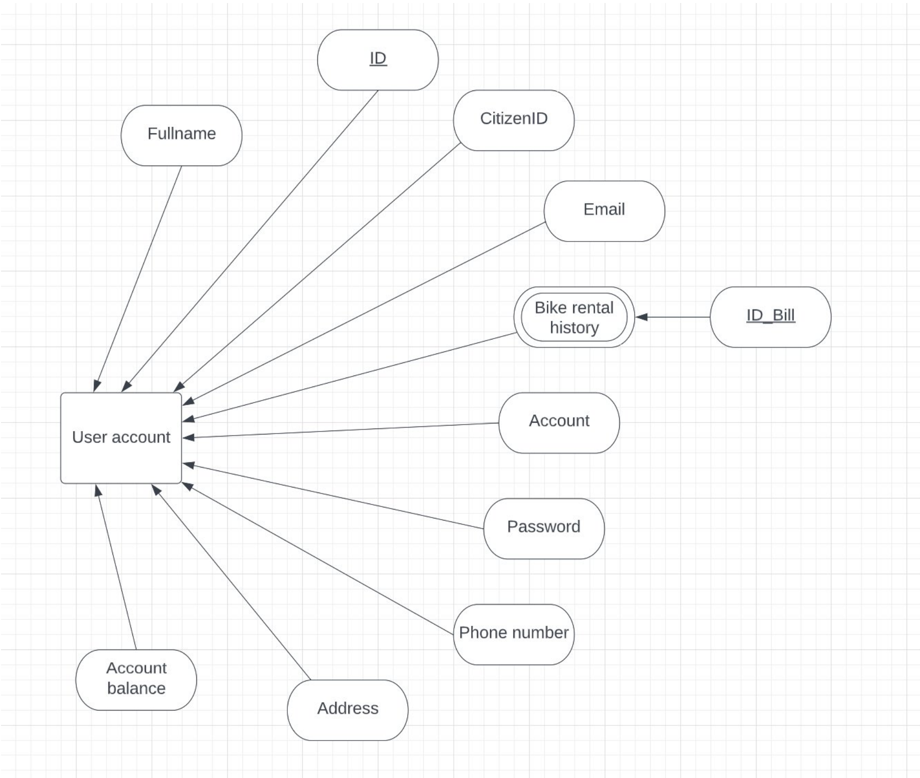


Figure 12.1: Entity User account

All account in the database, for every user (manager, employee, end user), have corresponding fields as shown in Figure 12.1.

- Bike and bike brand

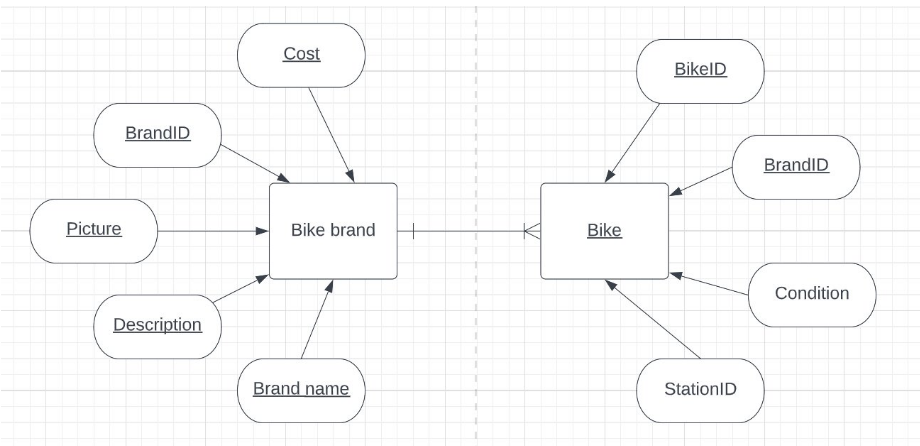


Figure 12.2: ERD of Entity Bike and Entity Bike brand

Bikes existing in the system will have corresponding fields, as shown in Figure 12.2. A bike brand has many bikes, but a bike has only one brand. This is the relationship between these two entities.

- Stations

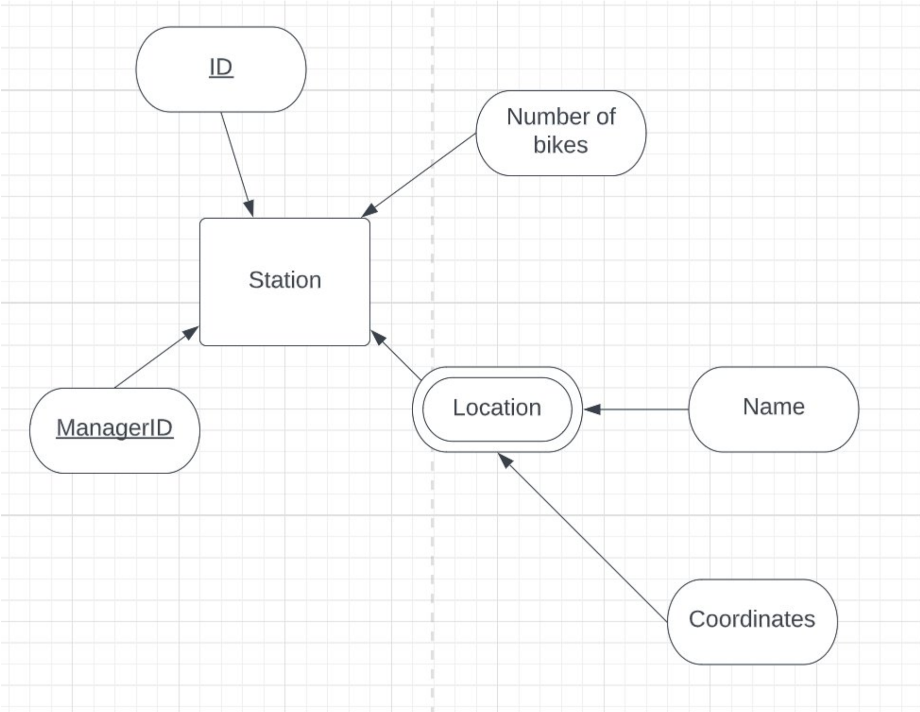


Figure 12.3: Entity Station

Entity Station will represent the stations on the system, containing pertinent information such as the location, the manager, and the current number of bikes available.

- Bike rental bill

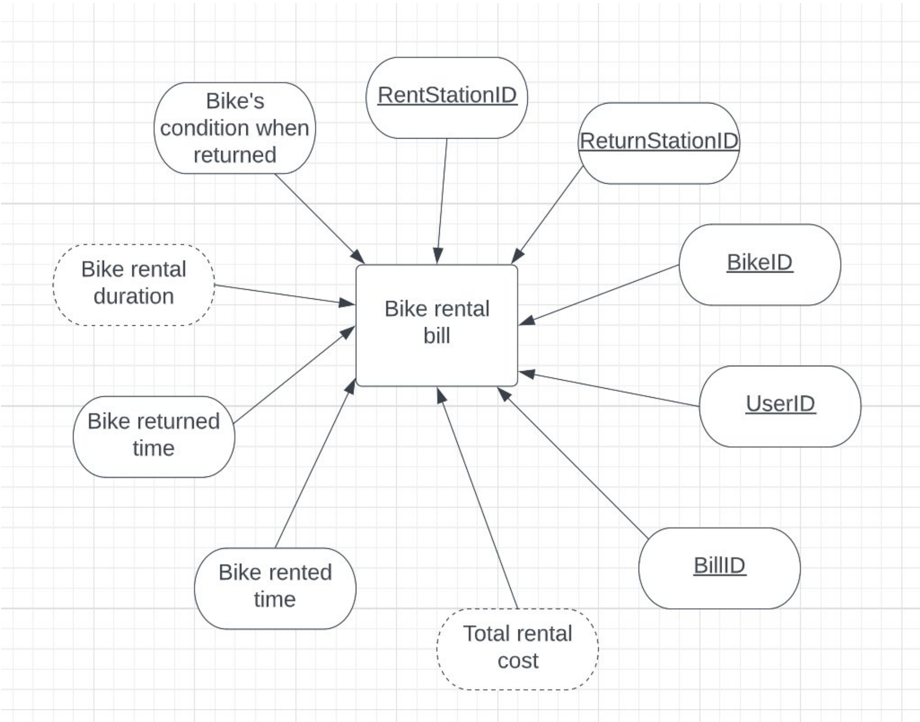


Figure 12.4: Entity Bike rental bill

Storing and processing bike rental bills needs information fields as shown in Figure 12.4.

5 Experiments

Experiment on a set of traffic velocity data for 207 intersections in the traffic system of a city in China. Link to the data here: https://github.com/lehaifeng/T-GCN/tree/master/data.
Data was measured over 168 hours for average speeds across 207 intersections. To keep things simple, the value at each point in time will be the average of all vehicles over 5 minutes on that traffic node. In the end, we have 2017 data points.
To convert this data into a time series, we use 13 5-minute periods in the past to predict 3 5-minute periods in the future. Divide the training set and testing set by the ratio of 0.8:0.2. Our data includes:

- 1598 samples for training
- 389 samples for testing

Data pre-processing: The model's activation functions are primarily sigmoid. In the range from -1 to 1, the sigmoid function's derivative is the most stable. It is advised to normalize the features in the input data X to the range from -1 to 1.

Frameworks used for implementation: sklearn, TensorFlow.
Parameters during training:

- Optimal Algorithm: Stochastic Gradient descent
- Learning rate: 1e-3
- Batch Size: 128
- Number of hidden units for BiLSTM layer: 64
- Number of epoch training: 1000 epochs
- Regularization factor: 0.0015

The results of the test set are measured on the following four matrices:

RMSE	5.112
MAE	3.560
R^2	0.565
Var	0.596

Conclusion: The algorithm gives very good results on the selected dataset. Graph Convolution Neural Network is a very good way to extract spatial features from data. Since the data is cyclical, the BiLSTM efficiently represents a temporal relationship.
Suggestion for improvement: To increase the predictive significance in terms of time, the next section will apply to the medium-term prediction problem (using the average velocity data of the previous two hours to predict the average velocity of the next two hours).
The speed information will be combined with the rules to deduce the status of the routes. From there, solve scheduling problems to coordinate traffic for the most convenient route.

6 Related Work

For a variety of reasons, including traffic congestion and environmental issues, many nations are growing more interested in BSSs. BSSs have thus been the subject of numerous studies. We categorize the pertinent BSS literature into three main categories in the section that follows. The analysis of data pertaining to bikes is included in the first category. A thorough analysis of the relevant data is needed in order to develop and manage a BSS effectively[6]–[10]. Data envelopment analysis (DEA), for instance, was used by Caggiani and colleagues[8] to assess the relative effectiveness of stations. The relative effectiveness of decision-making units can be assessed using the DEA model, which is based on linear programming[8]. Their findings are useful for organizing and overseeing BSSs. Yao and coworkers [7] also examined the network structures of BSSs. Stations were modeled as nodes, movements as edges, and the quantity of movements as edge weight in this case. They then observed the network properties and carried out several analyses, such as radiation distance, community structure, out- and in-degree, out- and in-strength.

The spatiotemporal biking patterns in BSSs were examined by Xiaolu Zhou[6]. They used hierarchical clustering to successfully analyze spatiotemporal patterns. Data from Chicago's Divvy BSS for the years 2013 and 2014 were used as the dataset. Additionally, Li and others[10] thought about how to handle massive amounts of BSS data with an effective bike usage representation. The dimensionality of the bike usage patterns was reduced using a discrete wavelet transform, and the reduced data were then clustered using k-means clustering. Decision-makers may gain useful insight into BSSs from the final clustering results.

In the second category, a prediction model is built. Effective bike rental and return prediction models have been developed using a variety of machine learning-based techniques [11]–[19]. Long short-term memory (LSTM)-based models, for instance, have been used in earlier studies [14], [16] because LSTM can capture long-term dependencies between data. Two layer LSTM models were employed to forecast bike demand in an earlier study[14]. These models' output was made up of both rentals and returns. The data from the Citi Bike System were also used as evaluation data in another study[14]. These statistics apply to the Citi Bike stations in Jersey City and New York City. An LSTM-based model was suggested by Sardinha and others [16] to take advantage of the characteristics of both historical and future contexts. In this case, the historical context was covered in the first layer, and the prospective context was covered in the second layer. Using the public BSS data from Lisbon, they experimentally proved the reliability of their system.

Deep learning models have recently been applied to traffic prediction. Liu[20] provided a thorough overview of the most recent deep learning-based methods for predicting traffic from mobility data. The prediction of traffic makes extensive use of convolutional neural networks (CNN). Meng[21] proposed PCNN, which only aims to take temporal dependencies into account, for short-term traffic congestion prediction. Ma [22] utilized CNN to predict traffic speed using a 2D time-space matrix. Since the trajectories are simply and linearly arranged, there may be a loss of spatial information between the trajectories. To extract spatial data for traffic flow prediction, Zhang [23] proposed a residual CNN called ST-ResNet based on a grid map. However, they disregarded the traffic flows' temporal tendency.

It is well known that LSTM [24]performs well when handling time series data. Unfortunately, Yu [25] did not take into account spatial information when applying deep LSTM to a time sequence in order to extract detailed information and predict traffic flows and accidents. Additionally, LSTM is appropriate for making traffic predictions for each road in a small area. However, both temporal and spatial factors have an impact on the traffic situation along a particular road segment in a road network.

Traditional machine learning models have been used in numerous other approaches [13], [18], [19] to predict bike demand. For instance, five forecasting models, including CUBIST, regularized random forest, classification and regression tree, K-nearest neighbors, and conditional inference tree models, were assessed to forecast bike demand in a prior study [19]. A log-log regression model for bike prediction was tested in a different study. Moreover, Gast and colleagues [26] attempted to model bike availability in a station using a queuing model, and Liu and Pelechrinis[13] adopted a Skellam regression model to estimate excess demand, for instance, how many users attempted to rent a bike from an empty station.

Although the methods attempted to create good BSS models, and the models are helpful for bike rebalancing and BSS planning, these earlier studies did not directly address the bike rebalancing issue. As a result, in order to increase the BSSs' usage effectiveness, we concentrate on the bike rebalancing problem in this paper.

The final category entails correctly rebalancing bikes to increase user convenience. The bike rebalancing issue has been examined in numerous prior studies[5], [8], [27]–[32]. For instance, Dell Amico and colleagues[33] used a stochastic programming model to resolve the two-stage problem of rebalancing a bicycle. Additionally, exact algorithms and heuristic algorithms have both been proposed as solutions to this issue [33]. An operator-based relocation method and a user-based relocation method were assessed in a different study[27]. By using an incentive, this user-based relocation technique rebalances bikes in accordance with BSS users. This approach has a low cost, but the rebalancing process is ineffective. The operator-based approach also employs operators, like relocation managers. This approach is very efficient but also quite expensive. A user-based relocation method based on the incentive concept was proposed by Cheng and others[5]. To maximize incentive prices, they suggested a bidding model-based incentive mechanism. By simulating each station's survival time and rebalancing the system as an optimization problem in a network, Chiariotti and colleagues[29] created a dynamic rebalancing technique. Here, they sought to increase user satisfaction while keeping rebalancing costs to a minimum.

7 Conclusion

In this paper, we have employed a software to automate the bike rental system and the T-GCN model to forecast traffic congestion. The software is built with a simple and easy-to-use interface, focusing on the main functions serving both managers and employees in the matter of managing automatic bike rental stations, thereby optimizing time and convenience for users. Managers can also view statistics and update the sales of every brand. The prediction problem is modelized using the macroscopic model with the data-driven approach for short-term prediction. With the T-GCN model, we use the GCN to extract the spatial feature; on the other hand, the BiLSTM architecture is used to extract the temporal feature. The algorithm produces excellent results for the chosen dataset.

References

[1] Y. Yao, Y. Zhang, L. Tian, N. Zhou, Z. Li, and M. Wang, “Analysis of

network structure of urban bike-sharing system: A case study based on real-time data of a public bicycle system," *Sustainability*, vol. 11, no. 19, p. 5425, 2019.

[2] C. H. Lee, J. W. Lee, and Y. J. Jung, "Practical method to improve usage efficiency of bike-sharing systems," in *ETRI Journal*, Apr. 2022, vol. 44, no. 2, pp. 244–259. doi: 10.4218/etrij.2021-0408.

[3] I. Frade and A. Ribeiro, "Bicycle Sharing Systems Demand," *Procedia Soc Behav Sci*, vol. 111, pp. 518–527, 2014, doi: https://doi.org/10.1016/j.sbspro.2014.01.085.

[4] S. Shaheen, E. Martin, and A. Cohen, "Public bikesharing and modal shift behavior: a comparative study of early bikesharing systems in North America," 2013.

[5] Y. Cheng, J. Wang, and Y. Wang, "A user-based bike rebalancing strategy for free-floating bike sharing systems: A bidding model," *Transp Res E Logist Transp Rev*, vol. 154, p. 102438, 2021.

[6] X. Zhou, "Understanding spatiotemporal patterns of biking behavior by analyzing massive bike sharing data in Chicago," *PLoS One*, vol. 10, no. 10, p. e0137922, 2015.

[7] Y. Yao, Y. Zhang, L. Tian, N. Zhou, Z. Li, and M. Wang, "Analysis of network structure of urban bike-sharing system: A case study based on real-time data of a public bicycle system," *Sustainability*, vol. 11, no. 19, p. 5425, 2019.

[8] L. Caggiani, R. Camporeale, Z. Hamidi, and C. Zhao, "Evaluating the efficiency of bike-sharing stations with data envelopment analysis," *Sustainability*, vol. 13, no. 2, p. 881, 2021.

[9] Y. Feng, R. C. Affonso, and M. Zolghadri, "Analysis of bike sharing system by clustering: the Vélib'case," *IFac-Papersonline*, vol. 50, no. 1, pp. 12422–12427, 2017.

[10] D. Li, Y. Zhao, and Y. Li, "Time-series representation and clustering approaches for sharing bike usage mining," *IEEE Access*, vol. 7, pp. 177856–177863, 2019.

[11] P. Hulot, D. Aloise, and S. D. Jena, "Towards station-level demand prediction for effective rebalancing in bike-sharing systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 378–386.

[12] G. Cantelmo, R. Kucharski, and C. Antoniou, "Low-dimensional model for bike-sharing demand forecasting that explicitly accounts for weather data," *Transp Res Rec*, vol. 2674, no. 8, pp. 132–144, 2020.

[13] X. Liu and K. Pelechrinis, "Excess demand prediction for bike sharing

systems," *PLoS One*, vol. 16, no. 6, p. e0252894, 2021.

[14] Y. Pan, R. C. Zheng, J. Zhang, and X. Yao, "Predicting bike sharing demand using recurrent neural networks," *Procedia Comput Sci*, vol. 147, pp. 562–566, 2019.

[15] A. A. Ramesh, S. P. Nagisetti, N. Sridhar, K. Avery, and D. Bein, "Station-level demand prediction for bike-sharing system," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021, pp. 916–921.

[16] C. Sardinha, A. C. Finamore, and R. Henriques, "Context-aware demand prediction in bike sharing systems: Incorporating spatial, meteorological and calendrical context," *arXiv preprint arXiv:2105.01125*, 2021.

[17] D. Singhvi *et al.*, "Predicting bike usage for new york city's bike sharing system," in *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, 2015.

[18] D. Singhvi *et al.*, "Predicting bike usage for new york city's bike sharing system," in *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, 2015.

[19] S. VE and Y. Cho, "A rule-based model for Seoul Bike sharing demand prediction using weather data," *Eur J Remote Sens*, vol. 53, no. sup1, pp. 166–183, 2020.

[20] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Netw*, vol. 32, no. 4, pp. 40–46, 2018.

[21] M. Chen, X. Yu, and Y. Liu, "PCNN: Deep convolutional networks for short-term traffic congestion prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3550–3559, 2018.

[22] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[23] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif Intell*, vol. 259, pp. 147–166, 2018.

[24] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 2018, pp. 1394–1401.

[25] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning:

A generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM international Conference on Data Mining*, 2017, pp. 777–785.
[26] N. Gast, G. Massonnet, D. Reijsbergen, and M. Tribastone, "Probabilistic forecasts of bike-sharing systems for journey planning," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 703–712.
[27] S. Reiss and K. Bogenberger, "A Relocation Strategy for Munich's Bike Sharing System: Combining an operator-based and a user-based Scheme," *Transportation Research Procedia*, vol. 22, pp. 105–114, 2017.
[28] B. Legros, "Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station," *Eur J Oper Res*, vol. 272, no. 2, pp. 740–753, 2019.
[29] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A dynamic approach to rebalancing bike-sharing systems," *Sensors*, vol. 18, no. 2, p. 512, 2018.
[30] S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Dynamic repositioning to reduce lost demand in bike sharing systems," *Journal of Artificial Intelligence Research*, vol. 58, pp. 387–430, 2017.
[31] C.-H. Lee, J.-W. Lee, Y.-J. Jung, and I.-Y. Cho, "Relocation Framework to Improve the Usage Efficiency of Bicycle-sharing Systems," in *2022 24th International Conference on Advanced Communication Technology (ICACT)*, 2022, pp. 424–428.
[32] P. Leclaire and F. Couffin, "Method for static rebalancing of a bike sharing system," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1561–1566, 2018.
[33] M. Dell'Amico, M. Iori, S. Novellani, and A. Subramanian, "The bike sharing rebalancing problem with stochastic demands," *Transportation research part B: methodological*, vol. 118, pp. 362–380, 2018.