Real-Time Analysis and Prediction System on Rail Transit Passenger Flow Based on Deep Learning

1st Xujun CHE
School of Science
Zhejiang University of Science & Technology
Hangzhou, China
chexj0000@163.com

2nd Gang CEN
School of Information and Electronic Engineering
Zhejiang University of Science & Technology
Hangzhou, China
gcen@163.com

3rd Shuhui WU
School of Science
Zhejiang University of Science & Technology
Hangzhou, China
s.wu@zust.edu.cn

4th Jiaming GU
School of Information and Electronic Engineering
Zhejiang University of Science & Technology
Hangzhou, China
gujm1004@163.com

5th Keying ZHU
School of Information and Electronic Engineering
Zhejiang University of Science & Technology
Hangzhou, China
z_keying1004@163.com

***Abstract*—With the rapid development of urban rail transit, rail transit plays an important role in alleviating city congestion. In recent years, with the passenger flow increasing there is a huge pressure on passenger flow management. To handle this problem, we propose a novel system to provide real-time statistics and predictions of passenger flow, which is based on big data technology and deep learning technology. What is more, the passenger flow is visualized efficiently in this system. It can provide the refined passenger flow information so that people could make more rational decisions in terms of operation and planning, to deploy contingency plans to avoid the emergency situation, and to integrate passenger flow analysis with train production, scheduling and operation to achieve cost reduction and efficiency enhancement.**

Keywords—rail transit, passenger flow, deep learning, big data

I. Introduction

With the rapid development of technological revolution and industrial transformation, China's traditional urban rail transit construction pattern has an undergone tremendous change. On March 12, 2020, China Association of Metros (CAMET) published and officially implemented the *Development Outline of Smart Urban Rail in China's Urban Rail Transit* to promote the development and evolution of smart urban rail with the mission of "transportation power and urban rail responsibility" [1]. Just one year later, the *Hangzhou Comprehensive Transportation Special Plan (2021-2035)* issued by the Hangzhou Municipal People's Government in September 2021. It shows that Hangzhou has included all the stations that have not been constructed in the third phase of rail transit and the stations in the last phase of the fourth phase into the scope of TOD integrated development, and vigorously advance the TOD integrated development and construction process. Thus, in the future, the job coverage rate of the population within 800 meters of the rail station will reach 45% [2], which means that the regional orbital density and the passenger flow density in Hangzhou will rise dramatically in the future. As a result of the iterative update of the city, the high volume and high density in piece area will face a huge challenge in the rail management.

Due to the advantages of large transportation volume, high punctuality rate and fast speed, the number of passengers in the metro keeps increasing. In the mean time, the travel data of passengers becomes larger and larger. For the metro managing, when decision makers face the difficulties of passenger flow management, they may not fully grasp the actual information of passenger flow distribution attributable to the unreasonable presentation of information, or fail to make adjustments in advance due to emergency fluctuations of the passenger flow. Metro stations usually are public places with dense crowds. Once the peaking of passenger flow exceeds the saturation number without in-time adjustment, it could cause the crowded stampede accident, which likely causes casualties [3]. To overcome this issue, an effective visual analysis and prediction system on passenger flow is necessary and important.

As the scale of metro travel data constantly increases, the traditional calculation methods can no longer meet the needs of today's large-scale passenger flow calculation. In order to process and analyze such huge data efficiently and precisely, big data processing technologies came into being. In fact, it has reached to a relatively mature stage, with the emergence of many well-known big data parallel processing platform such as Dryad [4], Hadoop [5], Spark [6], etc. These platforms not only solve the problem of insufficient single-machine resources but also have high computing efficiency.

For the passenger flow prediction, there are many different models proposed, including the early Kalman filter, exponential smoothing, autoregressive integrated moving average model (ARIMA), random forest, and the commonly used deep learning, such as convolutional neural network (CNN) [7], long short-term memory (LSTM) [8], gated recurrent unit (GRU) [9], etc. The accuracy of the models has reached a high level.

In this paper, we propose a novel system for multi-domain decision-making groups, with all-round analysis and highly user-friendly operation logic. It is implemented by big data technology and deep learning technology, which could provide users with real-time passenger flow statistics and predictions in the form of visualization. This could be a good solution for traffic managers to monitor the trend of passenger flow in order to prevent accidents.

II. System Design

The main functions of the real-time analysis and prediction system on rail transit passenger flow are gathering statistics and predicting the future passenger flows. The system firstly receives real-time passenger card swiping data from the Automatic Fare Collection (AFC) system of the metro, secondly uses big data technology to calculate real-time passenger flow, and thirdly displays it to users in the form of data visualization charts. At the same time, the system will save the passenger flow calculated in real time for afterwards analysis of historical passenger flow. The statistics and predictions in this system include four types of passenger flow: total flow, line flow, station flow and section flow. The overall design of the system is shown in Figure 1.

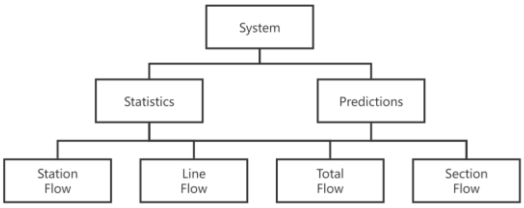

Figure 1: Overall design of the system

Among them, the total flow, the line flow and the station flow can be calculated directly through the passengers' inbound and outbound information. But the section flow is different. The section flow represents the passenger flow through the section between two stations. Its calculation requires the complete path of the passenger travel process, not just the inbound and outbound information [10]. Since the metro card swiping data only includes the records of passengers entering and exiting the station, but not the stations that passengers pass through before leaving the station, the complete route of the passenger travel process cannot be determined. This makes it difficult to calculate the precise value of section flow.

The system uses the Bellman-Ford algorithm [11] to obtain the shortest path for each origin-destination, which is these passengers' virtual path. In other words, it is assumed that each passenger uses the shortest path to reach their destination. The path information obtained by the above algorithm will be used to calculate the estimated value of section flow.

Through the area chart of station flow, line flow, and total flow provided by this system, it is possible to accurately grasp the passenger flow trend of a station, a line, or the entire city. At the same time, the system displays the passenger flow of every station with the sizes of the bubble, and the passenger flow of every section with the thicknesses of the lines on the metro map. Thus, the spatial distribution of passenger flow can be visualized directly among all stations and among all sections. The user interface design of the system is shown in Figure 2.

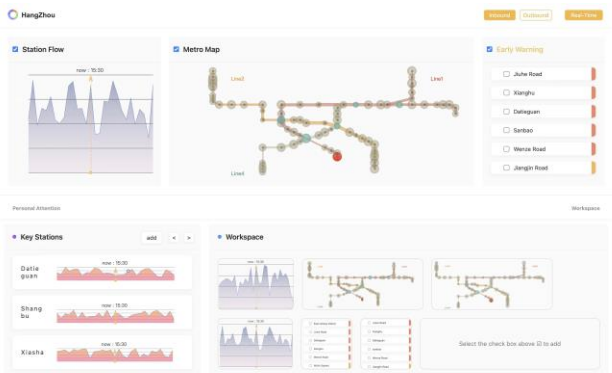

Figure 2: User interface design

III. Prediction Models

This system provides the function of passenger flow prediction. According to the different features of different types of passenger flow, the system adopts three different deep learning-based models to predict them.

A. *Total Flow and Line Flow*

The system uses the GRU-based Seq2Seq [12] model with attention mechanism [13] to predict total flow and line flow. The Seq2Seq model is a network composed of an Encoder and a Decoder. Its input is a sequence and its output is also a sequence, which meets the needs of the system to predict multiple time steps. Our model adopts two GRUs as Encoder and Decoder, respectively. Afterwards, an attention mechanism is introduced to improve the performance of the model. Such a structure can avoid the problem of error accumulation and thus achieves better performance under multi-step prediction.

B. *Section Flow*

Due to the huge number of metro sections, a large amount of parameters will be generated if the model is directly established. If so, the calculation efficiency will be quite low. Hence, our system adopts low-rank matrix factorization to solve this problem [14]. As shown in Figure 3, for the section flow matrix $\mathbf{A} \in \mathbb{R}^{n \times T}$ with n sections and T time steps, it can be factorized into the form of multiplying two matrices: $\mathbf{A} \approx \mathbf{S}\mathbf{T}^T$, where $\mathbf{S} \in \mathbb{R}^{n \times k}$ is the latent sectional embedding matrix and $\mathbf{T} \in \mathbb{R}^{T \times k}$ is the latent temporal embedding matrix. In order to get the optimal $\mathbf{S}$ and $\mathbf{T}$ we can solve:

$$\underset{\mathbf{S},\mathbf{T}}{\operatorname{argmin}} \sum_{i=1}^{n}\sum_{t=1}^{T} (\mathbf{A}_{it} - \mathbf{S}_i \mathbf{T}_t^T)^2 + \lambda_{\mathbf{S}} \|\mathbf{S}\|_F^2 + \lambda_{\mathbf{T}} \|\mathbf{T}\|_F^2 . \quad (1)$$

This optimization problem can be solved by alternating least squares or gradient descent. Then the system just needs to predict the temporal embedding matrix in the same way as total flow and line flow. Hence, the section flow prediction result can be obtained as $\mathbf{A}_{new} = \mathbf{S}\mathbf{T}_{\text{new}}^T$.

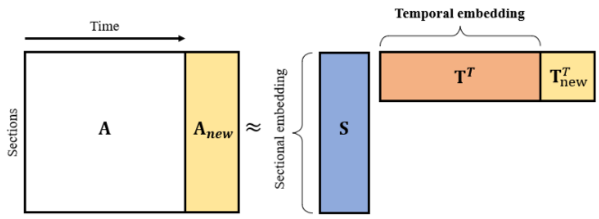

Figure 3: Matrix factorization for section flow

C. *Station Flow*

In order to make full use of the links and relative location relationships between stations, the system uses the T-GCN model [15] to predict station flow. This model combines the advantages of both GCN and GRU, using GCN to capture the spatial dependency of passenger flow between different stations, and then GRU to capture the temporal dependency of passenger flow, resulting in the final prediction result, which achieves the comprehensive use of spatiotemporal information and improves the accuracy of the prediction result.

IV. Technical Implementation

This system is constructed based on deep learning and big data technology. Its technical implementation is divided into four parts: client, server, network communication and prediction models.

A. Client

The client is programmed by TypeScript. Based on JavaScript, TypeScript adds a series of type checking functions and provides generic programming, which reduces the possibility of errors in the development process and greatly improves the reliability of the system. The interface is rendered by React. Using React's declarative paradigm, applications can be easily described. The reusability brought by its componentized development reduces development workload and improves code quality, and its virtual DOM feature minimizes page repainting, providing support for frequent real-time rendering of the system.

The client uses the open-source visualization chart library ECharts to draw diversified data visualization charts according to the characteristics of each statistical dimension. Echarts has rich chart types and high flexibility and extensibility. It can easily create all kinds of charts needed in this system and customize the styles, so that we can present suitable and beautiful charts for users.

B. Server

The system uses Kafka as a message queue, the Hadoop Distributed File System (HDFS) as a data storage, Spark as a big data computing engine, Redis as a cache, and Django as a service.

- ***Kafka*** is a distributed, high-throughput message queue system [16] that enables real-time messaging between different parts of the system. Kafka realizes the decoupling between different applications or between different parts of the same application, reduces unnecessary dependencies, and plays the role of peak shaving, avoiding the adverse impact of instantaneous traffic on the availability of the system.
- ***HDFS*** is a highly fault-tolerant distributed file system that provides efficient data management and access. The system uses it for the storage of historical passenger flow.
- ***Spark*** is adopted as the big data computing engine of the system. For easier and more efficient use of it, we use Spark SQL [17], a module of Spark that provides a better API called Data Frame. Structured Streaming is a high-level streaming API built on Spark SQL [18], and we use it to calculate real-time passenger flow. In such a framework, we can achieve a high performance.
- ***Redis*** is an in-memory non-relational database with very fast reading and writing speed, suitable for use as a cache. The system uses it to store real-time passenger flow over a period of time. This data will be used for prediction and deleted when not needed. In addition, by storing the queried historical traffic in Redis, the results can be retrieved directly from the cache when querying again without repeated calculation, which greatly improves the query speed and reduces the pressure on the server, and improves the operating efficiency of the system.
- ***Django*** is a web framework developed by Python, which can integrate with Redis, GraphQL, WebSocket and other components well.

C. Communication

During historical analysis, the client communicates with the server through the GraphQL API, via HTTP requests. GraphQL is a novel API query language that allows clients to get exactly the data they need, reducing data redundancy and request redundancy, increasing communication efficiency, and reducing the need for major changes, improving development efficiency.

During real-time analysis, the client communicates with the server via WebSocket, a protocol that supports persistent connections between the client and the server, which enables the server to actively push real-time passenger flow to the client.

D. Prediction Models

This system uses PyTorch to build the prediction models described in Section III. PyTorch is a very popular machine learning framework among researchers due to its ease of use. It allows us to build models more easily.

V. System Operation Process

As shown in Figure 4, the system obtains the passenger swiping card data constantly from the data source through Kafka, using Spark Structured Streaming to aggregate the data, using the deep learning models to predict future passenger flow, and communicating with the client via WebSocket. Finally, the statistical results and prediction results are presented to the user. At the same time, the system always saves the calculated real-time passenger flow to HDFS as the data of historical passenger flow for future calculation of historical statistics. For all the passenger card swiping data generated before the system is put into use, the system uses Spark to calculate the passenger flow every 10 minutes, then stores it in the distributed file system, and then uses Spark again to calculate the passenger flow in units of days through these data. This ensures data consistency in HDFS.

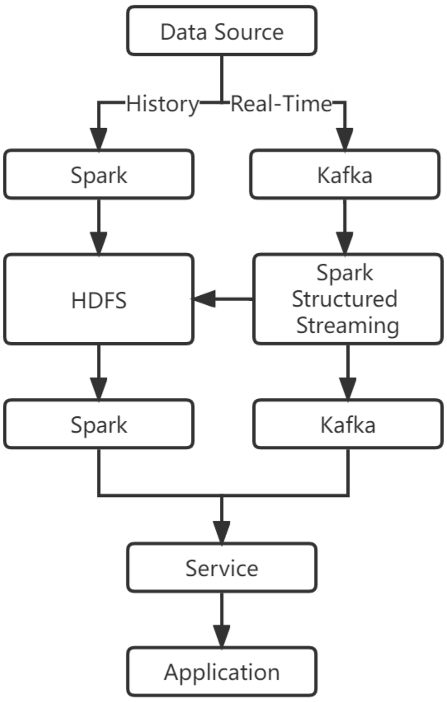

Figure 4: System operation process

VI. Features and Advantages

A. *Efficiency*

The system uses Spark, HDFS, Kafka and other technologies for distributed computing and data storage, with high computing efficiency, and various types of passenger flows can be quickly calculated.

B. *Real-Time*

Relying on high efficiency, the system can calculate the real-time passenger flow in time, so that users can see the real-time passenger flow statistical charts, analyze them, and make decisions accordingly.

C. *Accuracy*

According to the different features of different types of passenger flows, the system adopts different models to predict them, which has higher accuracy and provides users with a reliable decision-making basis.

D. *Intuitiveness*

The system provides area charts of station flow, line flow, and total flow, enabling users to understand the corresponding passenger flow trends. Furthermore, the system displays station flow and section flow in the form of bubble size and line thickness in the metro map, so that users can intuitively see the spatial distribution of passenger flow.

VII. Conclusion

On the basis of big data technology and deep learning technology, we proposed a system to make statistics and real-time predictions of metro passenger flow based on real-time swipe card data. The system adopted an intuitive and efficient visualization to provide a reliable data guarantee on in metro management as well as reduce the occurrence of accidents. Furthermore, it could also provide guidance for precautions of metro operation risk, optimal personnel allocation, engineering investment and business layout.

Acknowledgment

This work is supported in part by the grants of Zhejiang Xinmiao Talents Program under No. 2021R415025 and the Innovation and Entrepreneurship Training Program for Chinese College Students under No. 202111057017.

References

[1] China Association of Metros, “Development outline of smart urban rail in China's urban rail,” *China Metros*, vol. 8, no. 4, pp. 8-23, 2020.

[2] Nanjing Institute of City & Transport Planning Company Limited and Hangzhou City Planning and Design Academy, “Hangzhou comprehensive transportation special planning,” 2021.

[3] L. Yan, W. Tong, D. Hui, and W. Zongzhi, “Research and application on risk assessment DEA model of crowd crushing and trampling accidents in subway stations,” *Procedia Engineering*, vol. 43, pp. 494–498, 2012.

[4] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, “Dryad: Distributed data-parallel programs from sequential building blocks,” in *Proceedings of the 2Nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, ser. EuroSys ’07. New York, NY, USA: ACM, 2007, pp. 59–72.

[5] M. Ghazi and D. Gangodkar, “Hadoop, MapReduce and HDFS: A developers perspective,” *Procedia Computer Science*, vol. 48, pp. 45-50, 2015.

[6] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster computing with working sets,” in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud’10, Berkeley, CA, USA, 2010.

[7] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Computation*, vol. 1, pp. 541-551, 1989.

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

[9] K. Cho, B. V. Merrienboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *Computer Science*, 2014.

[10] C. Chen, and G. Liu, “Research on the calculation method of section passenger flow volume of urban rail transit,” *Technology & Economy in Areas of Communications*, vol. 17, no. 4, pp. 43-46, 2015.

[11] R. Bellman, “On a routing problem,” *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

[14] H.-F. Yu, N. Rao, and I. S. Dhillon, “Temporal regularized matrix factorization for high-dimensional time series prediction,” in *Advances in Neural Information Processing Systems*, pp. 847–855, 2016.

[15] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, “T-GCN: A temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[16] J. Kreps, N. Narkhede, J. Rao *et al.*, “Kafka: A distributed messaging system for log processing,” in *Proceedings of the NetDB*, pp. 1–7, 2011.

[17] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi et al., “Spark SQL: Relational data processing in spark,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM, pp. 1383–1394, 2015.

[18] M. Armbrust, T. Das, J. Torres, B. Yavuz, S. Zhu, R. Xin, A. Ghodsi, I. Stoica and M. Zaharia, “Structured streaming: A declarative API for real-time applications in apache spark,” in *Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 601-613.