Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/86/20241571

The investigation of traffic flow prediction and optimization based on Spark

Yikang Mai[1,3], Meng Xu[2]

[1]Financial Mathematics, Beijing Normal University - Hong Kong Baptist University United International College, Zhuhai, Guangdong, China
[2]Data Science and Big Data Technology, Zhejiang Wanli University, Ningbo, Zhejiang, China

[3]r130018042@mail.uic.edu.cn

Abstract. Spark and flink have made great strides in big data processing in recent years. Although the current spark can process a large amount of data at the same time, it still has a lot of shortcomings in the transparency, credibility, and practicality of the model. This paper provides a comprehensive overview of how to tackle the performance bottlenecks, insufficient model interpretability, and lack of regional adaptability faced by Spark and Flink in big data processing. It discusses the introduction of interpretable algorithms such as SHAP and LIME to enhance the transparency and user trust of neural network models. Then it discusses how to combine time-aware transfer learning and Geographic Information Systems (GIS) technology to enhance the technology generalization and adaptability of Spark machine learning models. Time-aware transfer learning uses historical data's temporal evolution to ensure models perform well in new time periods or scenarios, while GIS technology enables more precise predictions and analyses based on geographical data, enhancing spatial adaptability. Lastly, the study explores hybrid processing strategies by integrating Apache Flink, Kafka Streams, and Spark batch processing frameworks. This approach not only facilitates efficient real-time data processing and detailed analysis but also enhances the model's flexibility and processing capabilities in complex data scenarios. By integrating these techniques, it is possible to improve the efficiency and effectiveness of big data processing frameworks in addressing complex real-world challenges, thereby advancing technology and application development in related fields.

Keywords: Traffic flow prediction, machine learning, spark

1. Introduction

Transportation is an indispensable part of people's daily life, which always affects people's lives and traffic efficiency. With the acceleration of urbanization, the transportation system is facing tremendous pressure. It leads to the problem of traffic congestion, frequent traffic accidents and low management efficiency. Benefit form artificial intelligence, big data and other technologies, traffic efficiency and security have been improved through real-time data processing and intelligent traffic management.

In recent years, the field of Artificial Intelligence (AI) has made significant progress, with various algorithms and technologies continuously emerging, driving innovation and development across multiple domains. For example, deep learning, as a representative method, has achieved breakthroughs

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

in image recognition and natural language processing. Models like AlexNet [1], VGG [2], and ResNet [3], have performed excellently in image classification competitions, significantly advancing computer vision technology. Additionally, reinforcement learning has also demonstrated its powerful capabilities in games and robotics, with DeepMind's AlphaGo using reinforcement learning to defeat top human Go players [4]. Generative Adversarial Networks (GANs) have also shown outstanding results in image generation and restoration, with seminal works including the original GAN paper [5] and subsequent improvements such as CycleGAN [6] and StyleGAN [7].

The applications of AI are extensive, covering fields like chemistry, biomedicine, and transportation. Especially for the transportation sector, the application of AI is even more widespread and in-depth. Many researchers are dedicated to using AI technology to solve traffic congestion and optimize traffic flow. For example, Karras, T et al. have developed a smart parking system using computer vision and machine learning technologies to detect parking space usage in real-time, aiding drivers in finding available parking spots and reducing congestion caused by searching for parking [7]. Ault used reinforcement learning to develop an intelligent traffic signal control system that dynamically adjusts signal timings based on real-time traffic data, reducing congestion [8]. Nichola Foster et al used deep learning models to analyze urban traffic data and predict traffic flow changes, helping city planners optimize traffic management [9].

Spark, as a powerful parallel processing framework, plays a crucial role in combining AI with transportation due to its outstanding speed and processing capabilities. Spark can significantly reduce model training time and improve data processing efficiency, making it the tool of choice for many researchers. Spark's excellence lies in its in-memory computation and distributed data processing capabilities. By loading data into memory, Spark can achieve computation speeds up to 100 times faster than traditional disk-based processing frameworks like Hadoop. Additionally, Spark's Resilient Distributed Datasets (RDDs) provide fault tolerance, making it more stable and reliable when processing large-scale data. Zaharia, M. have developed a traffic flow prediction model using the Spark framework, achieving efficient traffic flow prediction through large-scale data analysis and parallel computing [10]. Therefore, it is necessary to provide a comprehensive summary of this aspect.

This article will introduce the concepts and framework of Spark, such as Spark SQL Spark Streaming, as well as the specific application cases of the technologies used in intelligent transportation, elaborate some key details in depth. Subsequently, this paper will summarize the achievements of intelligent transportation system in practical application, discuss the existing shortcomings. Putting forward achievable solutions and improvement directions to provide reference for future research and practice, so as to realize the feasibility of intelligent transportation.

2. Method

2.1. Introduction of Spark

Spark is a big data parallel computing model based on the Resilient Distributed Dataset (RDD) [11]. RDD is a read-only data collection stored in memory, and it undergoes transformations like map, groupByKey, and reduceByKey to accomplish data transformations. Its computing architecture is depicted in Figure 1. This model reads data from HDFS, forms RDDs, performs a series of transformation operations, and then writes the results back to HDFS [12]. When performing iterative operations using RDDs, the key-value RDD format is commonly used, where the dataset elements appear in the form of (key, value) pairs. In recent years, there have been numerous studies on traffic prediction based on Spark.

2.2. Spark in Traffic

Li et al. [13] proposed a real-time traffic flow prediction method that combines Flink stream computing framework and big data platform. They used Kafka messaging system to collect data from traffic sensors on road sections. After Flink streaming preprocessing, the data is sent to an independent distributed big data cluster, achieving real-time capture, preprocessing, and diversion of traffic flow data. They also proposed a parallel training mode for deep learning models based on the Hadoop big data platform, which makes full use of big data resources and technologies to achieve maximum data parallelism. Data generated by multiple road sensors in a certain traffic section were used to train and test the model. The sliding window is used to automatically select the nearest historical data set for model training and prediction, pursuing streaming automation and real-time processing. While maintaining a certain degree of accuracy, a faster model than GPU training was explored to meet the requirements of real-time prediction. By adopting this method, real-time capture, storage, and modeling analysis of massive data generated by multiple sensors in actual traffic sections are achieved, enabling real-time prediction of traffic flow in this section.

Ren et al. [14] proposed a Spark-based big data model called Spark-CoRLTT for calculating road segment travel time. This method completes the calculation in a single job, avoiding additional disk data storage and saving on computation time. The Spark-CoRLTT calculation method proposes a mapping relationship from the serial calculation process of road segment travel time to Spark data transformation, as shown in Figure 1. Phase 1, grouping by license plate, is implemented by a map called First map; Phase 2, sorting by vehicle passing time, is implemented by groupByKey; Phase 3, segmenting vehicle trajectories by road segment, is implemented by another map called Second map; and Phase 4, grouping by road segment and calculating road segment travel time, is implemented by reduceByKey. The improved algorithm significantly improves computational performance when dealing with large data scales.

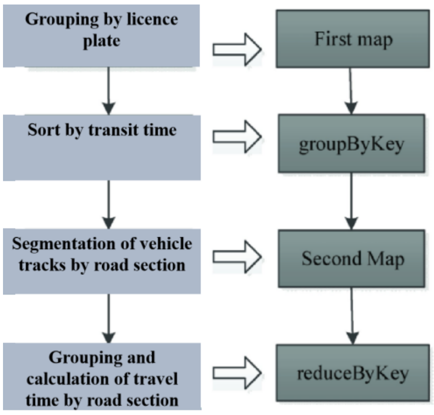
Grouping by licence plate
First map
Sort by transit time
groupByKey
Segmentation of vehicle tracks by road section
Second Map
Grouping and calculation of travel time by road section
reduceByKey

Figure 1. Mapping Relationships of the Spark-RLTT Algorithm (Photo/Picture credit: Original)

Gong et al. [15] designed a real-time traffic big data analysis and visualization platform utilizing big data geographic information system technology. The platform employs a distributed data storage structure combining HDFS and HBase to manage and store real-time traffic dynamic data. Leveraging the machine learning model Spark MLlib, it processes dynamic traffic information including image processing, traffic parameter extraction, traffic flow prediction analysis, and event detection. Through the Mapbox GL map rendering engine, the platform overlays and visually displays the results of traffic big data analysis and processing with map information. This enables functions such as real-time traffic condition analysis and display, short-term traffic prediction analysis, automatic detection of traffic events and congestion diffusion analysis, classification and statistics of historical traffic flow data, and transportation information services. This platform improves the storage, read-write efficiency, and analysis and processing capabilities of traffic big data, providing scientific and reliable decision support for urban traffic refined management.

The platform shown in Figure 2 utilizes Hadoop's HDFS to store different types of incoming raw data. HDFS supports distributed storage and adopts a master/slave architecture, providing complete redundant backup and fault recovery mechanisms to ensure the reliability of data read and write operations. HBase is a distributed open-source database built on the HDFS file system, supporting distributed storage with high reliability, high performance, scalability, and real-time read-write capabilities.

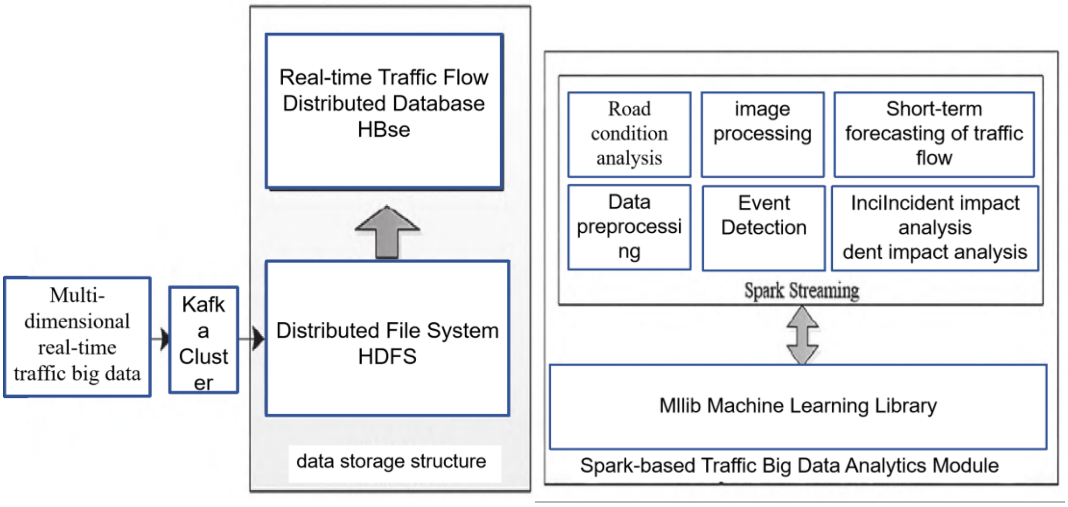
Multi-dimensional real-time traffic big data
Kafka Cluster
Real-time Traffic Flow Distributed Database HBse
Distributed File System HDFS
data storage structure
Road condition analysis
image processing
Short-term forecasting of traffic flow
Data preprocessing
Event Detection
IncilIncident impact analysis dent impact analysis
Spark Streaming
Mllib Machine Learning Library
Spark-based Traffic Big Data Analytics Module

Figure 2. Schematic Diagram of the Storage Structure and structure for Transportation Big Data (Photo/Picture credit: Original)

The traffic big data processing module utilizes Spark Streaming technology to provide various big data analysis capabilities, including traffic condition analysis, short-term traffic flow prediction, traffic event detection, traffic event impact analysis, and other functions.

3. Discussion

When dealing with big data and real-time data streams, although distributed computing frameworks such as Spark and Flink are powerful, they still face many challenges. Firstly, in terms of data processing and storage, for extremely large datasets, Spark may suffer from frequent disk I/O operations due to insufficient memory, thereby affecting performance. Additionally, the read-only nature of RDDs prevents dynamic updates, limiting their capability for real-time stream data processing. On the other hand, Flink may encounter state management and resource consumption issues when processing high-throughput data streams. Both frameworks require substantial computing resources to ensure real-time processing, which requires very high requirements for resource allocation and load balancing, otherwise it will lead to overloading of some nodes and affect the overall performance. Thus, whether using the Spark-CoRLTT model or Flink-based traffic prediction, the implementation and parallelization of complex algorithms can improve the performance, but also increase the difficulty of debugging and maintenance.

3.1. Interpretability

Spark is a big data parallel computing model based on the Resilient Distributed Dataset (RDD) [11]. RDD is a read-only data collection stored in memory, and it undergoes transformations like map, groupByKey, and reduceByKey to accomplish data transformations. Its computing architecture is depicted in Figure 1. This model reads data from HDFS, forms RDDs, performs a series of transformation operations, and then writes the results back to HDFS [12]. When performing iterative operations using RDDs, the key-value RDD format is commonly used, where the dataset elements appear in the form of (key, value) pairs. In recent years, there have been numerous studies on traffic prediction based on Spark.

In the fields of machine learning and artificial intelligence, interpretability of models has been a longstanding concern. Particularly with neural network models, which emulate the structure and function of the human brain, they excel in handling complex data patterns and prediction tasks. Despite their powerful capabilities in processing complex data patterns, the intricate connections and weight adjustments between layers involve a vast number of parameters and introduce non-linear activation functions, making their internal workings highly complex. This complexity often characterizes them as “black box” models. It’s difficult to intuitively understand and explain how each neuron in each layer and the connections and weights between them work together on the final prediction results. It is precisely because of the “black box” characteristics of neural networks that even if we predict the results, it is difficult to explain how the results are obtained.

Take traffic big data prediction as an example, where such models are commonly used to forecast traffic flow, congestion levels, and accident risks. However, erroneous predictions may lead decision-makers to implement ineffective traffic management strategies, such as improper signal timing, road closures, or traffic controls, these decisions based on mispredictions may not only fail to alleviate traffic problems, but may aggravate the risk of congestion accidents. Moreover, in emergencies like traffic accidents or unforeseen incidents, delays in accurate information from prediction models may hinder timely deployment of rescue teams and emergency resources. Therefore, enhancing the interpretability of neural network models, particularly in critical domains like traffic big data prediction, is crucially needed.

3.2. Applicability

The Machine Learning Library (MLlib) of Spark provides a rich array of algorithms and tools, enabling data scientists and machine learning developers to efficiently build, train, and evaluate models in a distributed environment. However, in the field of traffic big data applications, how to overcome regional differences—particularly between the North and the South, improving the generalizability and applicability of Spark-based machine learning models has become an urgent problem to be solved in the current traffic big data applications.

The northern and southern regions of China exhibit distinct contrasts in geography, climate, economy, and culture, all of which profoundly impact their transportation patterns. The northern region frequently faces icy and snowy road conditions in winter, thus favoring the use of heavy vehicles and trucks for transportation. In contrast, the southern region is more susceptible to rainfall and typhoons, leading to a higher usage rate of private cars and small trucks. Additionally, cultural differences, such as festival celebrations and travel habits, further exacerbate traffic flow fluctuations, ultimately affecting the road conditions in both regions.

These significant regional differences complicate and unevenly distribute the data collection process. To enhance the accuracy and applicability of models, researchers have to separately collect local data from the North and South, develop independent machine learning models tailored to these specific regions. This not only increases the complexity and cost of data collection but also requires a considerable amount of time and effort for model development, training, and tuning. More importantly, these independent models may be difficult to adapt to the new traffic due to the changing traffic conditions between regions, thus limiting the effectiveness and applicability of the models in broader scenarios.

3.3. Limited Real-time Capability

Spark excels in processing large-scale data due to its efficient in-memory computing model, extensive Application Programming Interface (API) and ecosystem, robust fault-tolerance, and compatibility with the Hadoop ecosystem, making it a preferred tool for big data tasks.

However, in the field of traffic big data which requires high real-time, spark’s batch processing framework needs to divide the data into multiple batches for processing, which causes the delay of data processing to some extent. Despite Spark Streaming simulating real-time processing through micro-batch processing, this latency can still delay the analysis of critical information in scenarios requiring millisecond response. For instance, in real-time traffic flow monitoring, newly arriving traffic data need to wait until the current batch processing completes, this may lead to delays of hundreds of milliseconds to seconds, which may hamper timely anomaly detection by traffic management authorities, restricting their ability to respond promptly, thus undermining traffic order maintenance. Similarly, in real-time accident alerts, if the warning information cannot be sent out in time because of the delay in processing, drivers and other road users will not be able to obtain the key avoidance information, which will greatly increase the risk and consequences of the accident.

4. Future prospects

4.1. Enhancing Model Interpretability

In order to solve the problem of the lack of interpretability of neural networks in traffic big data prediction, firstly, ensemble interpretability algorithms can be introduced. Using the Shapley Additive Explanations (SHAP) algorithm, the contributions of input features such as weather, time, and road conditions to the model's prediction results can be quantified. By analyzing the SHAP values, it is possible to clearly identify which factors have a significant impact on the prediction results, thus improving the interpretability of the model.In addition, hybrid models can be constructed to combine neural networks with expert systems and rule engines. By using expertise and rules from the transportation field, neural network predictions can be verified and adjusted to ensure that the results are both accurate and reliable. This comprehensive strategy not only enhances the interpretability of the model but also greatly improves the practicality and reliability of neural networks in traffic big data prediction, providing strong support for the sustainable development of intelligent transportation systems.

4.2. Improving Model Generalizability and Adaptability

In seeking ways to enhance the generalizability and adaptability of Spark-based machine learning models in transportation big data applications, the real-time and cyclical characteristics of traffic flow need to be considered comprehensively. Firstly, by incorporating time-aware functions into the transfer learning framework, models can more effectively adapt to the traffic pattern changes across different regions and seasons. Secondly, integrating Geographic Information Systems (GIS) and traffic simulation technologies can simulate the road network structures and traffic facilities of various regions, providing models with training data that more closely reflects real traffic conditions, thereby improving their accuracy. The combination of these strategies and technologies will significantly enhance the generalizability and adaptability of Spark machine learning models in traffic big data applications, providing smarter and more precise decision support for urban traffic management, and promoting the continuous development of intelligent transportation systems.

4.3. Hybrid Processing Strategies for Traffic Management

Hybrid processing strategies that combine real-time stream processing platforms like Apache Flink and Kafka Streams with the Spark batch processing framework can be used to enhance the efficiency and effectiveness of traffic management. Advanced frameworks can respond to data streams in milliseconds, effectively avoiding the latency issues inherent in traditional batch processing frameworks. By adopting a hybrid processing approach, the batch processing capabilities of Spark can be used for in-depth analysis of large-scale historical data, while real-time stream processing platforms handle newly generated traffic data in real time. In this way, traffic management departments can not only monitor traffic flow, detect anomalies, and respond quickly in real time, but also utilize historical data for long-term planning and decision-making. This comprehensive approach improves the efficiency and effectiveness of traffic management, effectively preventing traffic accidents.

5. Conclusion

With the advent of the big data era, traffic management systems face significant challenges in handling massive data and real-time data streams. Existing distributed computing frameworks like Spark and Flink, while helpful, still fall short in terms of performance, interpretability, and regional adaptability. These issues are critical as they impact the efficiency and accuracy of traffic management systems, limiting their application in complex urban environments. Addressing these challenges is crucial for reducing traffic congestion, decreasing accident rates, and improving overall urban efficiency and quality of life. To tackle this, we propose leveraging Spark and AI technologies. By introducing interpretability algorithms like SHAP and LIME, we enhance the transparency and trustworthiness of neural network models. Combining time-aware transfer learning and GIS technology improves model generalization and adaptability. Additionally, hybrid processing strategies that integrate Apache Flink, Kafka Streams, and Spark batch processing ensure efficient real-time data handling and detailed analysis. This approach not only enhances the performance and reliability of traffic management systems but also provides a robust technological foundation for smart city development. Future research will continue to optimize these methods to meet evolving real-world needs and challenges.

Authors contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Krizhevsky A, Sutskever I & Hinton GE 2012 ImageNet Classification with Deep Convolutional Neural Networks Advances in Neural Information Processing Systems vol 25 pp 1097-1105

[2] Simonyan K & Zisserman A 2015 Very Deep Convolutional Networks for Large-Scale Image Recognition International Conference on Learning Representations (ICLR)

[3] He K, Zhang X, Ren S & Sun J 2016 Deep Residual Learning for Image Recognition Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[4] Silver D, Huang A, Maddison CJ et al. 2016 Mastering the game of Go with deep neural networks and tree search Nature vol 529 pp 484-489

[5] Goodfellow I, Pouget-Abadie J, Mirza M et al. 2014 Generative Adversarial Nets Advances in Neural Information Processing Systems vol 27 pp 2672-2680

[6] Zhu JY, Park T, Isola P & Efros AA 2017 Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks Proceedings of the IEEE International Conference on Computer Vision (ICCV)

[7] Karras T, Laine S & Aila T 2019 A Style-Based Generator Architecture for Generative Adversarial Networks Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

[8] Ault A 2017 AI for traffic control: Reinforcement learning vs. traditional methods Retrieved from MIT News

[9] Intel 2024 Intel's intelligent traffic management system Retrieved from Intel's research blog (n.d.)

[10] Zaharia M, Chowdhury M, Das T et al. 2012 Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation (NSDI)

[11] Liu Y et al 2024 Online learning resource recommendation based on parallelized spectral clustering algorithm based on Spark platform Journal of Jinan University (Natural Science Edition) pp 1-5

[12] Ning Y, Chen JL, Luo M et al. 2024 Design and implementation of tourism big data analysis platform based on SpringBoot+Spark+Vue Wireless Internet Technology vol 21 (07) pp 60-67

[13] Li XH et al 2024 Real-time Traffic Flow Prediction Based on Stream Computing and Big Data Platform Computer Engineering and Design vol 45 (02) pp 553-561

[14] Ren G et al. 2023 A Calculation Method for Road Segment Travel Time Based on the Spark Model Wireless Internet Technology vol 20 (22) pp 101-107

[15] Gong XL et al 2023 Research on Real-time Traffic Big Data Analysis and Visualized Geographic Information Platform Journal of Guizhou Police College vol 35 (04) pp 77-83