

Article

Not peer-reviewed version

---

# Real-Time Data Processing in Smart Transportation Using Apache Spark

---

Nambua Ladslaus Mnyone and [Noor Ul Amin](#)\*

Posted Date: 17 March 2026

doi: 10.20944/preprints202603.1261.v1

Keywords: smart transportation; real-time data processing; Apache Spark; big data analytics; traffic monitoring



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Real-Time Data Processing in Smart Transportation Using Apache Spark

Nambua Ladslaus Mnyone and Noor Ul Amin \*

Taylor's University, No. 1, Jalan Taylor's, 47500 Subang Jaya, Selangor, Malaysia; nooraminnawab@gmail.com

## Abstract

Smart transportation systems generate large amounts of data from many sources such as GPS devices, traffic cameras, IoT sensors, mobile applications, and public transport systems. This data is produced continuously and needs to be processed quickly to support efficient traffic management and better transportation services. Traditional data processing systems are often slow when handling large and fast data streams, which makes real-time traffic analysis difficult. This paper discusses the use of Apache Spark for real-time data processing in smart transportation systems. Apache Spark is a big data processing framework that can analyze large volumes of data quickly using distributed computing and in-memory processing. The paper explains different transportation data sources and the challenges in collecting and integrating such data. It also describes the architecture and components of Apache Spark, including the driver program, cluster manager, executors, resilient distributed datasets, and Spark streaming.

**Keywords:** smart transportation; real-time data processing; Apache Spark; big data analytics; traffic monitoring

---

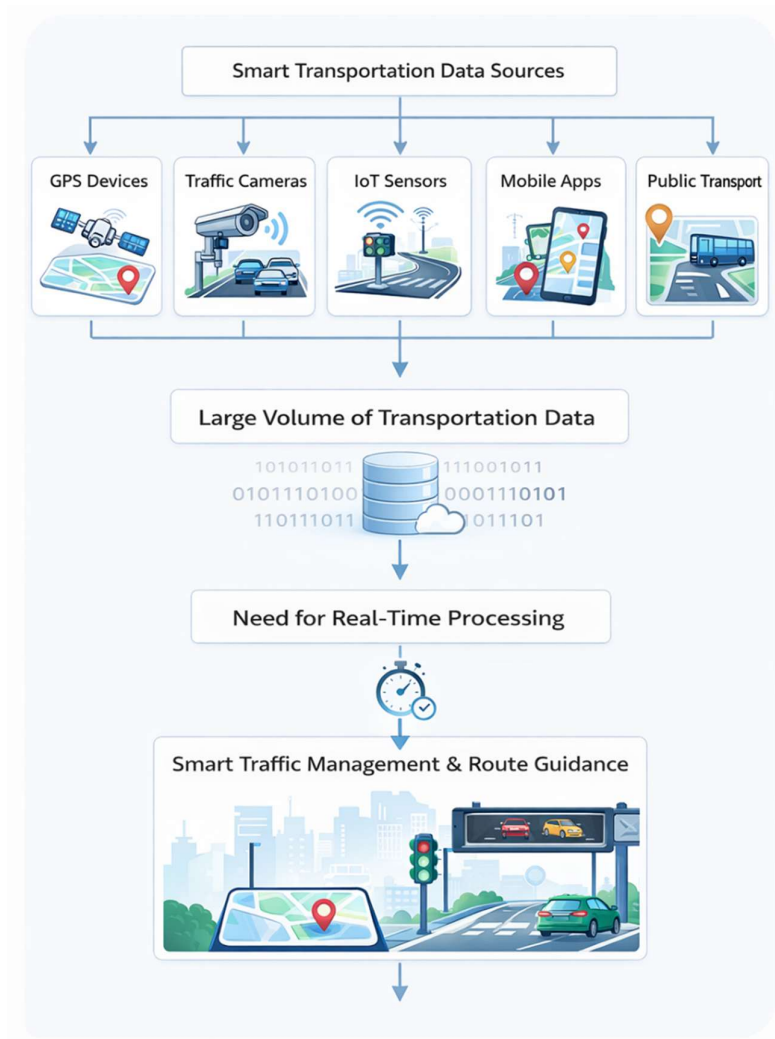
## Introduction

Today many cities use smart transportation systems [1]. These systems collect a lot of data from different sources such as GPS in vehicles, traffic cameras, road sensors, and mobile applications[2–4]. Every second new data is produced about vehicle locations, traffic speed, and road conditions. Because cities are growing, the amount of transportation data is also growing very fast [5].

This data needs to be processed quickly. If traffic information is processed slowly, drivers may receive updates too late[6,7]. For example, if an accident happens on a road, the system must detect it quickly and warn drivers. Real-time data processing helps traffic systems respond immediately and improve travel time and safety[8,9].

Traditional data processing systems are slow when handling very large amounts of data. Many of them store data on disks and process it step by step. This can take a long time, especially when the data is coming continuously from many sensors and devices [10,11].

Apache Spark is a modern big data framework that can process data much faster. It can analyze large datasets in memory and can also process streaming data in real time. Because of its speed and scalability, Apache Spark is suitable for analyzing transportation data and supporting smart traffic management systems [12].



**Figure 1.** Smart Transportation Data Sources.

## Data Sources and Collection

Data sources and collection for transportation systems involves the gathering of information from different mediums and technologies in order to analyze travel trends and improve services. There are various data sources in smart transportation systems and they are explained as follows:

Data sources used in smart transportation systems and their relevance:

- GPS Data from Vehicles and Smartphones:

Global Positioning System (GPS) data provides real-time information on vehicle locations, speeds and routes [13]. The information obtained can help identify popular routes and times of heavy traffic by showing how people move around cities. Current motor vehicles are often equipped with internal GPS system, and smartphones also add more data points, providing a thorough coverage of the urban transportation patterns. In addition to that the development of navigation applications, traffic congestion management, and route improvement are all enhanced by this data. For example, GPS data is used in navigational apps such as Waze and Google Maps to bring about real-time route suggestions and traffic updates. In an article of traffic management using real-time data [14,15]it discusses how real-time traffic data from GPS and other sources is used to improve traffic management.

- Internet of Things (IoT) Sensors:

Internet of Things (IoT) Sensors are devices that connect to the internet and collect data from their environment. IoT devices are such as traffic cameras, road sensors, and environmental monitors. Road sensors record the speed of vehicles, the vehicle types, and road conditions while traffic cameras track the number of vehicles and identify different incidents happening on the transportation infrastructures[16–18]. Environmental sensors monitor weather and air quality, which affects policies and strategies for transportation. Due to the integration of IoT devices, intelligent systems for traffic management that can adapt to changing conditions can be developed or made, making transportation efficient and reliable. [19] discussed on how an IoT-enhanced traffic light system implemented on Iraq's streets significantly improved the movement of traffic, anticipated congestion, and enhanced overall safety[20].

- Ride Sharing Platforms:

Large volumes of data about ride requests, trip durations, routes, and user preferences are produced by companies such as Uber, Bolt, Grab, Lyft and others [21]. Analyzing this data offers understandings of travel demand patterns, peak usage hours, and locations with few public transportation suppliers[22]. Urban planners who want to improve public transport networks and lessen traffic congestion may find this information quite helpful. For example, knowing the demand for ride-sharing can aid in creating specific areas for pickup and drop-off and improve public transport services in those areas, which will improve traffic flow. One of the analyses used include the uber data analysis made by [23,24] which aims at providing insights into ride-sharing usage patterns by analyzing trip data.

- Social Media Data:

Users of social media platforms like Facebook and Twitter exchange information in real time regarding accidents, traffic, and their experiences using public transportation. By assessing this data, it is possible to determine accident prone areas, identify public perspective, and get quick feedback on transportation services. As an example, unexpected increases in tweets regarding delays on a certain bus route may alert transportation officials to conduct an investigation and act quickly. According to [25], open platforms like SDK and API are used to gather public comments from popular social network platforms like microblogs. Additionally, using data from Twitter, Wanichayapong et al. created a system for collecting and classifying traffic statistics[26,27]. In order to display traffic information, Endarnoto et al. created an Android mobile application [28] and a Twitter traffic information gathering system.

- Public Transport Data:

Information from ticketing systems, train timetables and bus schedules provide details about how public transport operates. This comprises information on revenue tracking, passenger load analysis, and schedule enforcement. The development of different applications from outside developers that give users access to transit information such as transit app, google maps, moovit, onebusaway and others are made easier by the General Transit Feed Specification (GTFS), which is a common established format for public transportation schedules and related geographical information. Developers may develop applications that notify passengers of arrival times, service changes, and the best routes by having access to such common data[29].

## Challenges in Data Collection and Integration

Data collection and integration are very important components of current of modern data-driven decision-making, but they have different difficulties or challenges. There can be many barriers in the way of collecting data from multiple sources and then arranging it into an accessible and useful format, which makes it more difficult to obtain useful information. Organizations that depend on data to guide their operations and strategy need to understand these difficulties. Here are a few important issues:

- Data Heterogeneity:

Different data sources often have different formats and structures which makes it difficult to integrate or combine them easily. These data formats, types, and quality differ due to the variety of or different data sources. Thus, the establishment of standards is necessary to ensure the compatibility and consistency when combining data from the different data sources such as GPS units, Internet of Things sensors, social media platforms, and public transportation data. For example, public transport data is frequently schedule-based, whereas GPS data might provide continuous location updates. To properly combine and analyze these datasets, complex algorithms are required. Hence creating an overall overview of the transportation system requires addressing data heterogeneity. [30] Musa discussed the challenges facing integrated transport system in Malaysia where the difficulty integration of data from different sources can be witnessed.

- Data Volume:

The amount of data that is being created or generated by today's transportation systems is very huge. Processing and analyzing such enormous data in real time [31,32] is very difficult and it requires computational power and smart data management strategies or techniques. Detailed data is generated by different sources for example different vehicles provide detailed data of their location and speed frequently. In order to avoid overloading communication systems and exceeding computational capabilities, it is important that such data is collected efficiently.

- Data Accuracy:

Data gathered from various sources such GPS data, IoT sensors, and transport data can sometimes be incomplete, outdated, or noisy. For example, reports on social media such as twitter might be inaccurate or exaggerated, while GPS signals may experience multiple path problems when passing through areas such as urban passageways or tunnels[33]. Hence it becomes a challenge to obtain accurate data at all times, thus different techniques to improve data accuracy such as data validation methods, comparing data from several sources , and the use of machine learning to filter out errors need to be applied.

- Data Privacy:

Data collection raises various privacy concerns. This happens especially when it comes to data collection from personal devices like mobile phones[34–36]. Keeping the user data private and confidential is very important for maintaining public confidence and compliance with the legal regulations. Thus it is critical to implement effective data governance procedures, obtain informed consent or proper authorization before data collection , and make use of privacy-preserving technology in order to solve the different privacy issues[37–39].

Conclusively the development of intelligent transport systems depends critically on the incorporation of many data sources such as GPS data, IoT sensors, and transport data. Even if problems like data volume, data privacy, data accuracy or correctness, and data heterogeneity still exist, they are being addressed by the use of strict data management procedures and the continuous technology advancements[40–44].

## **Big Data Technologies for Smart Transportation**

Big data transportation technology refers to the use of advanced data analytics and processing tools to gather, to store, to analyze, and to utilize large volumes of data generated from various transportation systems such as busses, trains, vehicles and so much more. This technology uses data from sources like GPS devices, traffic cameras, sensors in vehicles, and mobile applications to improve transportation efficiency, safety, and sustainability. In other words smart transportation system are developing and advancing the way we move, making transportation more efficient, sustainable and user friendly. By using big data, these systems can optimize traffic flow, reduce congestion, and minimize environmental impact.

The key elements of big data technology system include data collection which is the gathering data from various sources, including IoT devices, social media and traffic management systems, data

processing by employing tools such as Apache spark and flink for real time processing and analytics to derive actionable insights. Data storage which is done by utilizing frame works like Hadoop to store large amounts of structured and unstructured data in a secure way, data analysis which is the analyzing the collected data from different sources to identify patterns, predict traffic conditions, optimize routes, and improve the overall transportation services. Lastly data visualization which uses dashboards and visualization tools to present data insights in a user-friendly manner for decision makers.

## Recommended Big Data Tools and Frameworks

### 1. Apache Spark

This is a framework for distributed storage and processing of large data sets.it provides real time data processing and analytics capabilities, making it ideal for traffic management and route optimization, it has sparks in memory processing that allows faster data analysis compared to traditional disk-based processing (zaharia etel,2020).

Ideal use cases include batch processing, which is suited for big data tasks like log analysis and data warehousing, and real-time analytics, which is used for fraud detection with streamed data, as well as analyzing sensor data or social media feeds. Apache Spark is used for machine learning as well. Among other things, Spark works well for creating recommendation systems and loss prediction. Additionally, it facilitates interactive data exploration by using Spark SQL for data discovery and ad hoc analysis. Additionally, predictive models for traffic flow, congestion prediction, and maintenance scheduling can be created using Spark's machine learning libraries (MLlib).



Figure 2. Apache Spark Logo.

## Spark Architecture and Components



Figure 3. Apache Spark Components.

### 1. Driver Program

The driver program is the central controller that starts and manages the execution of a spark application. This creates a spark context, which serves as the entry point to interact with sparks. The driver program is responsible for defining the spark job, such as processing traffic data or analyzing the crowd. It coordinates tasks in several activists' nodes in a cluster, divides large computations into small tasks and distributes them efficiently. Once the processing is completed once, the driver program collects results from activists' nodes, which ensures a smooth and organized execution of the application.

## 2. Cluster Manager

The cluster manager is responsible for allocating resources and managing communication between various spark components. Spark supports several cluster managers (apache Spark, n.d.), including standalone mode, the inherent cluster manager of sparks, Yarn (yet another resource - negotiator) which is usually used in the Hadoop environment and Mesos which is a common-purpose cluster manager. Lastly Kubernetes that is a container-based cluster manager.

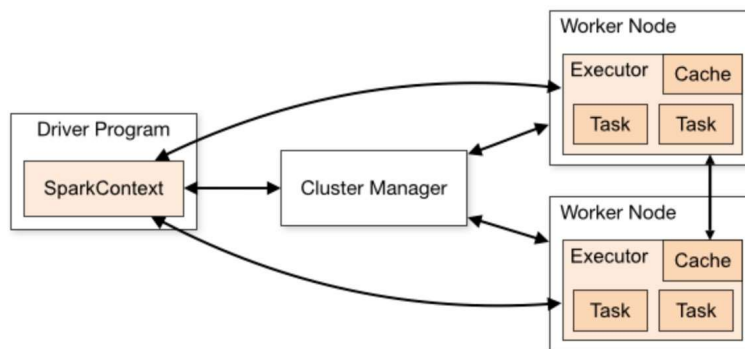


Figure 4. Role of Cluster Manager.

The role of the cluster manager involves spark applications, scheduling work, assigning CPU and memory resources, and monitoring the health of data nodes or slave nodes. The cluster manager also ensures the efficient job execution and balance in clusters. For example, in the traffic monitoring system, a spark job may require a large-scale data stream to process GPS device and IOT sensor. The cluster manager dynamically allocates computing resources, ensuring smooth data processing without system overload.

## 3. Executors

The executor workers are procedures or processes that run on cluster nodes and calculate the driver's program. Each spark application has its own set of executors, which are responsible for executing assigned tasks on divided data, intermediate calculations in memory are stored or are on discs for rapid processing, and return the final processed results to the driver. The executors work in parallel, which makes sparks highly efficient for large-scale data processing, the executors handle data exchange between nodes, facilitating distributed data processing (*Apache Spark Architecture: Concepts, Components, and Best Practices*, 2024). For example, in a traffic monitoring system, executors can process real-time speed data from vehicles, analyze patterns, and predict future crowded areas.

## 4. Resilient Distributed Dataset (RDD)

The resilient distributed dataset (RDD) is the main data structure of the spark, which enables fault-tolerant and parallel processing (Altexsoft, 2023). RDDs store data in many nodes, which ensure both reliability and performance.

The major features of RDD include partition data, which allows RDD to divide large datasets into small parts that can be processed concurrently. Another important feature is irreversibility this means that once an RDD is made, it cannot be modified, which ensures stability.

Additionally, RDD provides fault tolerance whereby if a node fails, it can reconnect the lost data using the information of the spark dynasty.

For example, in a traffic monitoring system, GPS data from various vehicles can be stored in RDD and divided by location. Spark can then process this data in parallel, simultaneously detect crowds in different city areas.

### 5. Streaming Module

Spark streaming is a real-time processing module that enables sparks to handle continuous currents of data. It divides streaming data into micro-batch and processes them using the same spark engine as batch jobs.

Major capabilities of spark streaming include the ability to ingest real-time data from sources such as Kafka, Floom or IOT sensor. Spark streaming processes data in real time hence allowing the detection of patterns, discrepancies, or events (*Apache Spark Architecture: Concepts, Components, and Best Practices, 2024*). In addition to that, it can be integrated with machine learning models for future stating analytics.

For example, in a traffic monitoring system, a spark streaming job continuously processes live data from traffic cameras and GPS devices. This may detect real-time congestion, accidents, or road closure, updating traffic control centers immediately.

### 6. MLLib (Machine Learning Library)

MLLIB is the inherent machine learning library of spark, designed for future analysis and adaptation tasks. It provides a series of algorithms that can be applied to various scenarios, including the prediction of traffic patterns, passage optimization and discrepancy.

For predicting traffic patterns, MLLIB may estimate the crowd based on historical data. In terms of route adaptation, it suggests the drivers with fastest or minimum crowded routes. Additionally, it plays an important role in detecting discrepancy by identifying abnormal traffic behaviors, such as accidents or sudden speed drop.

In a traffic monitoring system, MLLIB may analyze the trend of previous traffic to predict extreme congestion hours. It may recommend alternative to the navigation systems such as Google Maps or Veg, which increases the overall efficiency of the journey.

### 7. Spark SQL

Spark SQL is an Apache Spark module that is designed to process structured data (*Apache Spark Architecture: Concepts, Components, and Best Practices, 2024*). Spark sql enables users to run SQL queries with spark programs, facilitating spontaneous integration between SQL-based data analysis and large data processing. Spark SQL supports various data formats including JSON, Parquet, AVRO and ORC, which is a versatile tool for data handling (ayushjoshi599, 2020).

Spark SQL plays an important role in analyzing structured transport data, such as GPS data, traffic signal and vehicle telemetry. In a smart transport system, spark SQL can process historical and real-time data to increase decision making.

For example, in public transit, spark SQL may analyze the passenger travel pattern based on ticketing data and suggest route optimization to reduce congestion. This can process the live traffic feed to detect peak hours and adjust traffic signals dynamically for better flows.

Additionally, the spark SQL improves urban planning by analyzing larger datasets from many different transport sources, such as ride-sharing apps, buses and metro services.

By integrating with a database storing traffic and transport records, Spark SQL enhances future stating analytics, allowing city planners to design more efficient transport systems that delay and reduce carbon footprints.

### 8. Graphx

Graphx is the library of Apache spark for graph processing and calculation (ayushjoshi599, 2020). This enables users to model, analyze and manipulate graph-structured data efficiently. In a smart city, graphx can model the entire road network to find the most efficient path for public transport and emergency vehicles.

In a smart city, graphx can model the entire road network to find the most efficient path for public transport and emergency vehicles. It can run the shortest path algorithm to recommend the

fastest route in real time, reducing travel delays. For example, in a traffic management system, graphx may detect crowded areas and suggest alternative routes to improve overall mobility.

In addition, graphx can help prevent accidents by analyzing historical accident data and identifying high-risk intersections. It also supports ride-sharing and fleet management services by optimizing vehicle distribution on the basis of demand pattern.

### 9. Apache Spark Core

The Apache Spark Core is the fundamental execution engine of the Apache spark which provides essential functionality such as task scheduling, memory management, fault tolerance and distributed data processing. This enables in-memory computing, which greatly speeds up data analysis by reducing dependence on disc storage. Additionally, it allows integration with external storage systems such as HDFS, Amazon S3 and Apache Cassandra, which makes it adapt to various data sources(Gupta, 2024).

In a smart transport system, the spark core plays an important role in real-time traffic analysis and decision making. It processes large amounts of transport data by taking advantage of GPS signal, traffic sensor readings, and ride-sharing logs.

**Table 1.** Simplified Explanation on Spark Components.

Spark Component	Description
<b>Driver Program</b>	Manages and controls Spark applications, assigns tasks, and collects results.
<b>Cluster Manager</b>	Allocates resources, schedules tasks, and ensures efficient execution across nodes.
<b>Executors</b>	Perform the actual data processing by running assigned tasks in parallel.
<b>Resilient Distributed Dataset (RDD)</b>	A fault-tolerant, distributed data structure that enables parallel processing.
<b>Streaming Module</b>	Processes real-time data streams in micro-batches for immediate analysis.
<b>MLlib (Machine Learning Library)</b>	Provides built-in machine learning algorithms for predictive analysis.
<b>Spark SQL</b>	Allows querying and processing of structured data using SQL-like commands.
<b>GraphX</b>	Supports graph processing and analytics for network-based data.
<b>Spark Core</b>	The main engine that handles task scheduling, memory management, and fault tolerance.

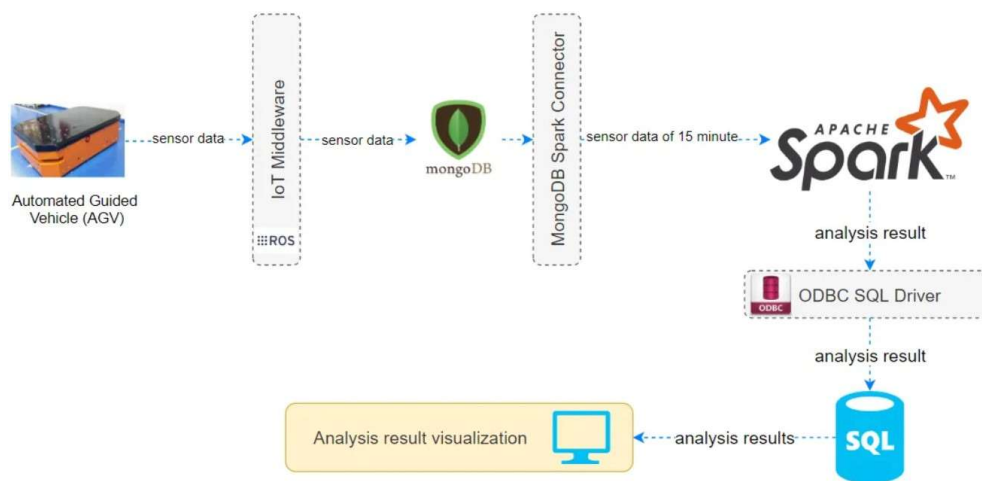


Figure 5. Apache Spark in Smart Transportation (Autonomous Vehicles).

## Justification for Using Apache Spark

### ○ Fast Processing

One of the biggest benefits of sparks is its in vast memory processing model, which allows it to be largely processed much faster than a traditional disc-based processing framework such as Hadoop MapReduce.

Instead of storing intermediate data on the disc after each computation phase, such as Hadoop, the spark holds data in RAM. This approach eliminates expensive disc i/o operations, which significantly reduces processing time. If the memory is inadequate, the spark efficiently manages the disc spill to the disc without the major performance recession.

This rapid processing capacity is particularly important for traffic monitoring. Traffic data from GPS devices, IOT sensors and cameras is constantly streaming in real time. If a system takes a long time to process this data, traffic alerts and congestion reports can be chronic and ineffective. With sparks, real-time traffic updates can be processed within the millisecond, which ensures rapid and more accurate decisions.

For example, a smart city traffic control system uses Apache sparks to analyze live data or real-time from thousands or even millions of sensors. If an accident occurs on a highway, the spark immediately detects the unusual crowd patterns and sends alert to the officers and navigation apps. This quick response enables time, reduces the risk of further accidents and delays.

### ○ Advanced Machine Learning Support

Advanced machine learning support is an important feature of Apache Spark which includes the MLLIB (machine learning library). This machine learning library provides powerful tools that are used to detect future analysis and discrepancy, making it a valuable resource for various applications including traffic management.

It is necessary to understand the traffic flow, as it is affected by various factors such as weather conditions, day time and special events. Machine Learning algorithm can analyze these factors to predict traffic patterns, detect discrepancies and optimize the routes using both historical and real-time data.

There are many major machine learning applications in traffic management. An important application is a prediction of traffic congestion. Machine learning models can analyze the previous or historical traffic data to identify the pattern, allowing the system to redeem alternative routes to predict and avoid delay.

Another important application is to detect accidents and discrepancy. Abnormal traffic behavior, such as sudden braking or uncertain movement, may indicate an accident. Machine Learning algorithms can automatically flag these potential accidents, alerting authorities to rapidly response.

The root optimization is also extended by the MLLIB of the spark. It can analyze traffic trends and suggest optimal travel routes for passengers, which helps in reducing fuel consumption and overall congestion at the time of travel.

For example, navigation systems such as Google Maps has the machine learning library of sparks integrated into it. This system continuously learns from past and live or real-time traffic data to recommend the fastest routes. If it detects a traffic jam, it can recreate drivers immediately to avoid delays, improving the overall efficiency of traffic management.

#### ○ Scalability

Scalability is an important aspect of managing transport data in Smart City Infrastructure, especially as the expansion of these systems and the amount of data increases rapidly. Spark scalability ensures that it can efficiently handle the increasing data load.

Spark obtains scalability by following a distributed computing model, where the data is divided into several nodes (servers) and processed in parallel. It uses dynamic resource allocation, which means that it can automatically be up or down on the basis of charge. Additionally, the RDD (flexible distributed dataset) of the sparks allows the dataset to be processed in many machines without slowing it.

This scalability is particularly important for traffic monitoring. Thousands of IOT sensors, GPS trackers and cameras are deployed in smart cities that generate large amounts of data. If the traffic monitoring system cannot score efficiently, the data bottleneck and processing may be delayed. Spark ensures that such as the number of data sources increases, the performance remains sharp and reliable.

For example, if a city expands its transport network by adding new smart traffic lights, vehicle sensors and public transit tracking systems, Spark can basically integrate these new data sources without the need for a major system overhaul. As a result, the city can manage its growing transport network without experiencing recession or system failures.

## Conclusion

Smart transportation systems produce a large amount of data from many sources such as GPS devices, traffic cameras, IoT sensors, mobile apps, and public transport systems. This data grows every second. Because of this, it is important to process the data quickly so that traffic conditions can be understood in real time.

Traditional data processing systems are often slow when dealing with very large and continuous data streams. This makes it difficult to respond quickly to traffic congestion, accidents, and changing road conditions.

Apache Spark provides a powerful solution for real-time transportation data processing. It can process large volumes of data quickly and supports distributed computing, streaming analytics, and machine learning. These features make Spark suitable for applications such as traffic monitoring, congestion prediction, and route optimization.

In the future, real-time big data systems will play an important role in improving transportation efficiency, reducing traffic delays, and supporting the development of smarter cities.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Oladimeji, D., Gupta, K., Kose, N. A., Gundogan, K., Ge, L., & Liang, F. (2023). Smart transportation: an overview of technologies and applications. *Sensors*, 23(8), 3880.

2. Fantin Irudaya Raj, E., & Appadurai, M. (2022). Internet of things-based smart transportation system for smart cities. In *Intelligent Systems for Social Good: Theory and Practice* (pp. 39-50). Singapore: Springer Nature Singapore.
3. Singh, B. (2023). Federated learning for envision future trajectory smart transport system for climate preservation and smart green planet: Insights into global governance and SDG-9 (Industry, Innovation and Infrastructure). *National Journal of Environmental Law*, 6(2), 6-17.
4. Elassy, M., Al-Hattab, M., Takruri, M., & Badawi, S. (2024). Intelligent transportation systems for sustainable smart cities. *Transportation Engineering*, 16, 100252.
5. Ray, S. K., Pawlikowski, K., & Sirisena, H. (2008, October). A fast MAC-layer handover for an IEEE 802.16 e-based WMAN. In *International Conference on Access Networks* (pp. 102-117). Berlin, Heidelberg: Springer Berlin Heidelberg.
6. Kunjir, S. N., Patil, S. S., Hingane, B. S., Pagariya, J. A., & Rashid, M. (2023, September). Managing smart urban transportation with the integration of big data analytic platform. In *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)* (Vol. 6, pp. 1807-1811). IEEE.
7. Chowdhury, M., Dey, K., & Apon, A. (Eds.). (2024). *Data analytics for intelligent transportation systems*. Elsevier.
8. Samaras, V., Daskapan, S., Ahmad, R., & Ray, S. K. (2014, November). An enterprise security architecture for accessing SaaS cloud services with BYOD. In *2014 Australasian Telecommunication Networks and Applications Conference (ATNAC)* (pp. 129-134). IEEE.
9. Niyazi, M., & Behnamian, J. (2023). Application of cloud computing and big data in three-stage dynamic modeling of disaster relief logistics and wounded transportation: a case study. *Environmental Science and Pollution Research*, 30(13), 38121-38140.
10. Ushakov, D., Dudukalov, E., Kozlova, E., & Shatila, K. (2022). The Internet of Things impact on smart public transportation. *Transportation Research Procedia*, 63, 2392-2400.
11. Dey, K., Ray, S., Bhattacharyya, P. K., Gangopadhyay, A., Bhasin, K. K., & Verma, R. D. (1985). Salicyldehyde 4-methoxybenzoylhydrazone and diacetylbis (4-methoxybenzoylhydrazone) as ligands for tin, lead and zirconium. *J. Indian Chem. Soc. (India)*, 62(11).
12. Dritsas, E., & Trigka, M. (2025). Applying machine learning on big data with Apache Spark. *IEEE Access*.
13. Anwesa Chaudhuri, A. C., & Sanjib Ray, S. R. (2015). Antiproliferative activity of phytochemicals present in aerial parts aqueous extract of *Ampelocissus latifolia* (Roxb.) Planch. on apical meristem cells.
14. Azeem, M., Abualsoud, B. M., & Priyadarshana, D. (2023). Mobile big data analytics using deep learning and Apache Spark. *Mesopotamian Journal of Big Data*, 2023, 16-28.
15. Alexakis, T., Peppes, N., Demestichas, K., & Adamopoulou, E. (2022). A distributed big data analytics architecture for vehicle sensor data. *Sensors*, 23(1), 357.
16. Ray, S. K., Sirisena, H., & Deka, D. (2013, October). LTE-Advanced handover: An orientation matching-based fast and reliable approach. In *38th annual IEEE conference on local computer networks* (pp. 280-283). IEEE.
17. Ushakov, D., Dudukalov, E., Kozlova, E., & Shatila, K. (2022). The Internet of Things impact on smart public transportation. *Transportation Research Procedia*, 63, 2392-2400.
18. Gul, O. M. (2022). Blockchain-enabled Internet of Things (IoTs) platforms for vehicle sensing and transportation monitoring. *Machine Learning, Blockchain Technologies and Big Data Analytics for IoTs: Methods, Technologies and Applications*, 351-373.
19. Fantin Irudaya Raj, E., & Appadurai, M. (2022). Internet of things-based smart transportation system for smart cities. In *Intelligent Systems for Social Good: Theory and Practice* (pp. 39-50). Singapore: Springer Nature Singapore.
20. Shah, I. A., Jhanjhi, N. Z., & Laraib, A. (2023). Cybersecurity and blockchain usage in contemporary business. In *Handbook of Research on Cybersecurity Issues and Challenges for Business and FinTech Applications* (pp. 49-64). IGI Global Scientific Publishing.
21. Younas, F. (2025). Factors influencing consumer choice in ride sharing platform: a study on Uber and Bolt in Helsinki.

22. Azeem, M., Ullah, A., Ashraf, H., Jhanjhi, N. Z., Humayun, M., Aljahdali, S., & Tabbakh, T. A. (2021). Fog-oriented secure and lightweight data aggregation in iomt. *IEEE Access*, 9, 111072-111082.
23. Ker, S. K., & Liu, S. (2022). Uber Freight: Assessment and Determination of Optimal Design Features for a Drop Trailer Service Offering and Network (Doctoral dissertation).
24. Lee, S., Abdullah, A., & Jhanjhi, N. Z. (2020). A review on honeypot-based botnet detection models for smart factory. *International Journal of Advanced Computer Science and Applications*, 11(6).
25. Agarwal, P., & Alam, M. (2020). Open service platforms for IoT. In *Internet of Things (IoT) Concepts and Applications* (pp. 43-59). Cham: Springer International Publishing.
26. Muzafar, S., & Jhanjhi, N. Z. (2020). Success stories of ICT implementation in Saudi Arabia. In *Employing Recent Technologies for Improved Digital Governance* (pp. 151-163). IGI Global Scientific Publishing.
27. Li, D., Zhang, Y., & Li, C. (2019). Mining public opinion on transportation systems based on social media data. *Sustainability*, 11(15), 4016.
28. Endarnoto, S. K., Pradipta, S., Nugroho, A. S., & Purnama, J. (2011, July). Traffic condition information extraction & visualization from social media twitter for android mobile application. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics* (pp. 1-4). IEEE.
29. Liu, D., Guo, J., Gu, Y., King, M., Han, L. D., & Brakewood, C. (2025). Analyzing Transit Systems Using General Transit Feed Specification (GTFS) by Generating Spatiotemporal Transit Networks. *Information*, 16(1), 24.
30. Musa, A. A., Malami, S. I., Alanazi, F., Ounaies, W., Alshammari, M., & Haruna, S. I. (2023). Sustainable traffic management for smart cities using internet-of-things-oriented intelligent transportation systems (ITS): challenges and recommendations. *Sustainability*, 15(13), 9859.
31. JingXuan, C., Tayyab, M., Muzammal, S. M., Jhanjhi, N. Z., Ray, S. K., & Ashfaq, F. (2024, November). Integrating AI with robotic process automation (RPA): advancing intelligent automation systems. In *2024 IEEE 29th Asia Pacific Conference on Communications (APCC)* (pp. 259-265). IEEE.
32. Abdulkarim, S., Mohktar, S., Babaji, N. A., Maisalati, M. S., Musa, M. K., & Pwara, K. E. (2021). Roles of road transport policy on economic development in federal capital territory abuja, nigeria: stakeholders perception. *Management*, 7(27), 179-197.
33. Abdulkarim, S., Mohktar, S., Babaji, N. A., Maisalati, M. S., Musa, M. K., & Pwara, K. E. (2021). Roles of road transport policy on economic development in federal capital territory abuja, nigeria: stakeholders perception. *Management*, 7(27), 179-197.
34. Javed, D., Jhanjhi, N. Z., Ashfaq, F., Khan, N. A., Das, S. R., & Singh, S. (2024, July). Student performance analysis to identify the students at risk of failure. In *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)* (pp. 1-6). IEEE.
35. Tang, L., & Na, S. (2021). Comparison of machine learning methods for ground settlement prediction with different tunneling datasets. *Journal of Rock Mechanics and Geotechnical Engineering*, 13(6), 1274-1289.
36. Bora, P. S., Sharma, S., Batra, I., Malik, A., & Ashfaq, F. (2024, July). Identification and classification of rare medicinal plants. In *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)* (pp. 1-6). IEEE.
37. Shreyas, S. K., & Dey, A. (2019). Application of soft computing techniques in tunnelling and underground excavations: state of the art and future prospects. *Innovative Infrastructure Solutions*, 4(1), 46.
38. Badu-Marfo, G., Farooq, B., & Patterson, Z. (2019). A perspective on the challenges and opportunities for privacy-aware big transportation data. *Journal of Big Data Analytics in Transportation*, 1(1), 1-23.
39. Miao, Y., Yang, Y., Li, X., Choo, K. K. R., Meng, X., & Deng, R. H. (2023). Comprehensive survey on privacy-preserving spatial data query in transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(12), 13603-13616.
40. Jabeen, T., Jabeen, I., Ashraf, H., Ullah, A., Jhanjhi, N. Z., Ghoniem, R. M., & Ray, S. K. (2023). Smart wireless sensor technology for healthcare monitoring system using cognitive radio networks. *Sensors*, 23(13), 6104.
41. Callegati, F., Campi, A., Melis, A., Prandini, M., & Zevenbergen, B. (2015). Privacy-preserving design of data processing systems in the public transport context. *Pacific Asia Journal of the Association for Information Systems*, 7(4), 4.

42. Singha, S., & Singha, R. (2023). Protecting data and privacy: cloud-based solutions for intelligent transportation applications. *Scalable computing: practice and experience*, 24(3), 257-276.
43. Sui, P., Li, X., & Bai, Y. (2017). A study of enhancing privacy for intelligent transportation systems: A correlation privacy model against moving preference attacks for location trajectory data. *IEEE Access*, 5, 24555-24567.
44. Ying, Z., Cao, S., Liu, X., Ma, Z., Ma, J., & Deng, R. H. (2022). PrivacySignal: Privacy-preserving traffic signal control for intelligent transportation system. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 16290-16303.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.