

Original Research Paper

# A Scalable Big Data Framework for Real-Time Traffic Monitoring System

<sup>1,2</sup>Wilfried Yves Hamilton Adoni, <sup>3,4</sup>Najib Ben Aoun, <sup>2</sup>Tarik Nahhal, <sup>3,5</sup>Moez Krichen, <sup>3</sup>Mohammed Y. Alzahrani and <sup>6</sup>Franck Kalala Mutombo

<sup>1</sup>Engineering School, International University of Casablanca, Casablanca, Morocco

<sup>2</sup>FDMS Research Unit, Hassan II University of Casablanca, Casablanca, Morocco

<sup>3</sup>College of Computer Science and Information Technology, Al Baha University, Al Baha, Saudi Arabia

<sup>4</sup>REGIM-Lab Research Groups in Intelligent Machines, University of Sfax, Sfax, Tunisia

<sup>5</sup>ReDCAD Laboratory, University of Sfax, Sfax, Tunisia

<sup>6</sup>Department of Mathematics, University of Lubumbashi, Lubumbashi, Democratic Republic of Congo

## Article history

Received: 02-06-2022

Revised: 31-07-2022

Accepted: 01-09-2022

Corresponding Author:

Najib Ben Aoun

College of Computer Science  
and Information Technology,  
Al Baha University, Saudi  
Arabia

Email: najib.benaoun@ieeee.org

In this study, a scalable and real-time intelligent transportation system based on a big data framework is presented. The proposed system allows for the use of existing data from road sensors to better understand traffic flow, and traveler behavior and increase road network performance. Our transportation system is designed to process large-scale stream data to analyze traffic events such as incidents, crashes, and congestion. The experiments performed on the public transportation modes of the city of Casablanca in Morocco reveal that the proposed system achieves a significant gain of time, gathers large-scale data from many road sensors, and is not expensive in terms of hardware resource consumption.

**Keywords:** Road Sensor, GPS Sensor, Intelligent Transportation System, Big Data, Smart City, Traffic Monitoring, Urban Mobility, Hadoop, IBM InfoSphere

## Introduction

Nowadays, the new technologies bring many benefits and the transport sector does not escape this rule Jabbar *et al.* (2021, 2020); Abbas *et al.* (2021). Thanks to the mass of data generated, new strategies can be developed to anticipate certain unforeseen events, making the systems even more efficient and reliable.

The exploitation of big data technology enables transportation systems to analyze a significant amount of stream data to prevent traffic jams, incidents, delays, or track given vehicles in real-time. Big data technology has been used for object detection Brahimi *et al.* (2017; 2018a), Event detection Aoun *et al.* (2011; 2014), and semantic segmentation Brahimi *et al.* (2018b; 2019) and had an impact on users' lifestyles. Indeed, the whole data of the traffic coming from several sources of information can be crossed for an effective decision-making aid concerning the mobility of the travelers.

Several projects and research works have been done in big data technologies; this new technology allows us to cross the boundaries of the traditional database management system. This allows us to face the challenges of massive data processing and encompasses the new data

management innovations for better performance and productivity growth of enterprises Manyika *et al.* (2011). Zheng *et al.* (2013) have done research on the evolution of air quality in overpopulated cities. By using big data real-time analytic tools, they provide information about air quality. By combining historical and real-time air data from existing monitor stations and weather stations, they provide derived information on real-time air pollution in the cities of Beijing and Shanghai. This study had a great impact on the environment by reducing the emission of greenhouse gases.

Our research was motivated by Adoni *et al.* (2017a,b, 2018); Biem *et al.* (2010a,b); Bouillet and Ranganathan (2010); Kitchin (2014) and made feasible by access to data from Casablanca's buses, tramways and GIS data of Morocco. The data originates from a variety of sources, including Global Positioning System (GPS) sensors, traffic control devices, and car embedded systems. Helpful information is provided to end-users in this area by integrating static data with mobility data. The volume and variety of traffic data, the veracity, and the velocity with which all traffic events are dealt in this area.

In this study, the main architecture of the proposed distributed transportation system is presented. The proposed system is based on a big data framework and allows the processing of historical and real-time traffic data. This method incorporates a wide range of technologies that may be used to handle several transportation issues, such as traffic prediction, regularization difficulties, the Traveling Salesman Problem (TSP), the shortest path problem, and so on. IBM big data technology is chosen as a solution, which includes Infosphere Big Insights, Infosphere Streams, and Infosphere Warehouse, among other platforms.

## Related Work

The big data concept attracts the most interest in transport because the quantity of information managed is too large and more complex Li *et al.* (2015); Kitchin (2014); Hao *et al.* (2015). Current intelligent transportation systems provide multi-modal services adapted for the community. Nevertheless, it becomes a perilous task when it is a request taking into account several mobility constraints. In work conducted by Toole *et al.* (2015); Dong *et al.* (2015), the authors used big data techniques to cross and analyze data from traffic flow and call detail records to predict the density of traffic flow on the road network.

In the case of road traffic, all data generated by passengers is progressively augmented by additional data collected in real-time by an expanding number of traffic sensors. Dodge and Kitchin (2007) have developed a transportation system that analyzes the movement of cars on the road network in real-time. They evaluate vehicle speeds and automatically apply fines for speeding violations by integrating data from road sensors. The police utilize this information to respond quickly to incidents and create a log file collecting all vehicle tracking records.

The notion of big data and machine learning were combined by Lécué *et al.* (2014) in the forecast of traffic incidents in Dublin, Ireland. Tests were performed to prove the capability of the system to collect and manage a large amount of historical sensor data and react in real time to all captured traffic events. In other words, Biem *et al.* (2010b, a) proposed an intelligent framework based on big data stream flow. The proposed system collects a large volume of data from many GPS sensors and combines them with road network data. Similarly, Bouillet and Ranganathan (2010) use the same method to generate a picture of traffic conditions using data from GPS sensors.

Recently, Adoni *et al.* (2017a) have introduced a MapReduce-based approach to analyze traffic stream

data to detect abnormal traffic events. Their approach consists of partitioning the log file of traffic events into a set of sub-events and then analyzing traffic events on each event block in a parallel way. Their proposed technique is very inspiring and achieves significant gain in terms of runtime complexity. They used a similar technique to propose a parallel and distributed version of the A\* pathfinding algorithm called MRA\* (MapReduce-A\*) Adoni *et al.* (2017b, 2018) which consists to combine the A\* algorithm with the MapReduce paradigm to compute the shortest path on the large-scale road network.

## Proposed Framework

The proposed system is built to handle massive amounts of data from the traffic flow. A method based on the 4V (Volume Velocity Variety Veracity) properties of big data is used. In this context, big data solutions are used because it is an integrated production platform and includes all Apache Hadoop ecosystems (MapReduce, Hive, Pig, Impala, Mahout, HBase, Sqoop, etc.). It helps to take on the challenges and the complexity of the MapReduce program allowing users with limited SQL knowledge to be able to manipulate a lot of unstructured data. The proposed system allows for the collection and analysis of petabytes of structured and unstructured traffic data, to build real-time predictive models of traffic flow to detect and react quickly to any traffic events.

## Architecture Overview

Figure 1 shows an architectural overview of our system and how it works in practice. The proposed architecture is based on Hadoop IBM Big Insights. The first layer is a stream layer, it contains all Stream Process Applications (SPA) that allow management data in motion. The processing of the traffic stream is composed of three phases. The first phase entails real-time traffic data processing. This involves getting data from embedded devices in automobiles and processing it. The data is consolidated and combined with data from road sensors in the second phase, allowing it to react quickly to incidents that disrupt traffic conditions. Vehicle location, incident detection, congestion or traffic jam detection, and vehicle delay are all activities conducted during this phase. Finally, the distributed file system is used to store the streaming data.

The second layer concerns data storage, which incorporates all Hadoop components that allow you to work with information stored in a distributed file system to create ad hoc analysis requests and prediction models across time intervals using analytic tools.

The third layer is based on a standard analytical model similar to an n-tier architecture. The analytical and predictive treatments are performed through Big Insights analytical tools (SPSS and RStudio). In the first step of this treatment, real-time data are acquired and combined with static data. This step consists to extract, transform and filter the traffic dataset (pre-processing). The second step concerns the statistical analysis to predict traffic conditions. Statistical models combined with the stream analytic process allow the real-time prediction of traffic. The final step provides visualization of the Key Performance Indicator (KPI), dashboard, and map view of analytical results. The end-user layer ensures the presentation of related traffic information. It is the end-users layer that allows one to see traffic information via the web console and smartphone equipped with a GPS sensor Krichen (2021).

### *Stream Layer*

Large data streams created in real-time are called data in motion, this provides for a fast response to congested road events as well as continuous communication with end-users. Streams programs are created that evaluate streaming data in real time. Several data sources feed information flows; to improve the design and execution of the streaming process, the data sources are categorized into two groups: (1) road sensors and (2) tramway checkpoints, as well as data from embedding GPS in buses. These smart objects also provide information on the mobility of passengers such as entrances/exits and peak hours.

The general framework of the traffic stream graph is depicted in Fig. 2, it consists of a collection of data sources linked to operators. Large amounts of unstructured data generated by a variety of sensors are supported by IBM Stream.

A common set of communication functions is used to access data from the sensors (socket). Many adapters are available in Infosphere Streams for gathering information via data transfer protocols. The first step is to figure out which communication protocol various sensors employ. Other types of exchange protocols can also be added if the platform does not support them. The second stage is to conduct real-time traffic data analysis. To do this, Infosphere Streams includes a variety of operators such as transformations, correlate, filter, annotate, aggregate, and so on. Streams link all of the operators, each operator connects with the others immediately through their input and output ports. The stream operator ends with the detection of abnormal events from all traffic data from the cluster as a whole.

Our stream layer is fully compatible and can be used for the design of the streaming process on the Apache

Flume NG. The operating mode of this platform is based on stream agents (Source, Memory Channel and Sink).

Two types of events are captured from traffic data: (1) Incident detection: Detect all vehicles with abnormal behavior (vehicle with speeding and accident between two or more vehicles); (2) Traffic congestion: Characterized by slow traffic speeds and an increased queue of vehicles. The operators used in the stream layer of our transportation system are:

1. File Source: The File Source operator is used to read data from the file and generates tuples as a result
2. TCP Source: The TCP Source operator is used for reading data from a TCP socket and generates tuples as a result
3. UDP Source: The UDP Source operator is used for reading data from a UDP socket and generates tuples as a result
4. File Sink: The File Sink operator is used for writing the output of tuples into local files or the distributed file system
5. Filter: The Filter operator is used for filtering tuples based on specific criteria
6. Custom: The Custom operator is used to define specific functions and logic clauses

Figure 3 shows the stream pipeline of fraud and accident detections. It consists of File Source and UDP/TCP Source, File Sink, Filter, and Custom operators. The filter operator filters all vehicle streams based on their speed while the custom operator implements the Incident Detection function. The principle of fraud detection consists of detecting all vehicles whose speed exceeds the speed limit. In case of collision between vehicles, the vehicles are assumed to be stopped on the roadway. Based on this assumption, all vehicles with zero speed are identified. The final step is to determine the crash location, it is made possible by getting all GPS positions (latitude and longitude) of the damaged vehicles.

Figure 4 shows the stream pipeline of congestion detection. It consists of File Source, File Sink, Aggregate, and Custom operators. The Aggregate operator allows to merge and count all vehicles by roadway. The custom operator implements a Congestion Detection function. The custom function takes as input the density, critical density, and jam density. The density represents the number of vehicles on roadway length at a given time.

The critical density is the highest density that can be supported under free flow. Jam density refers to extreme traffic density when traffic flow stops completely. In the case of congestion, the density is between the critical density and jam density while in the case of traffic jam, the density is greater than the jam density.

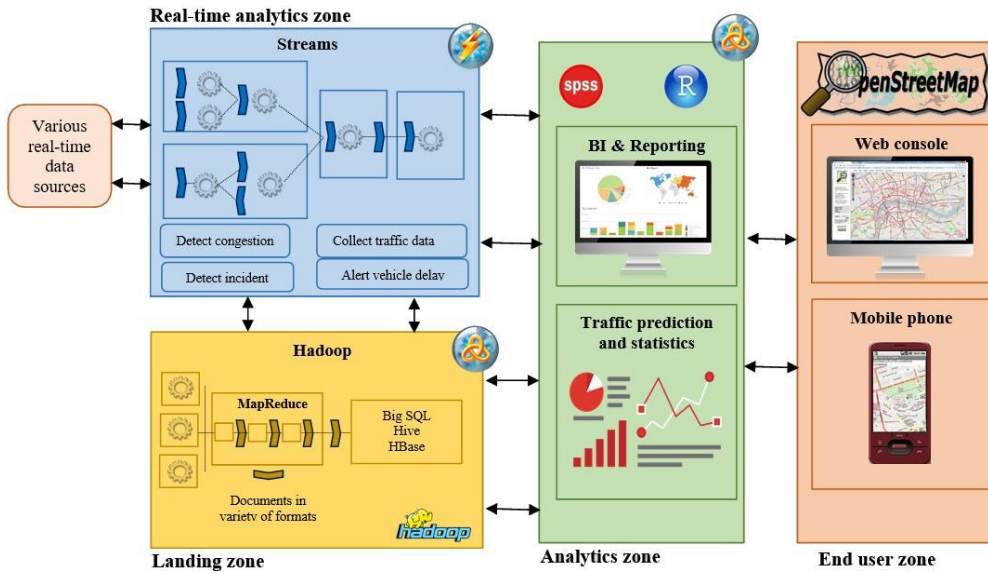


Fig. 1: Architecture of our intelligent transportation system

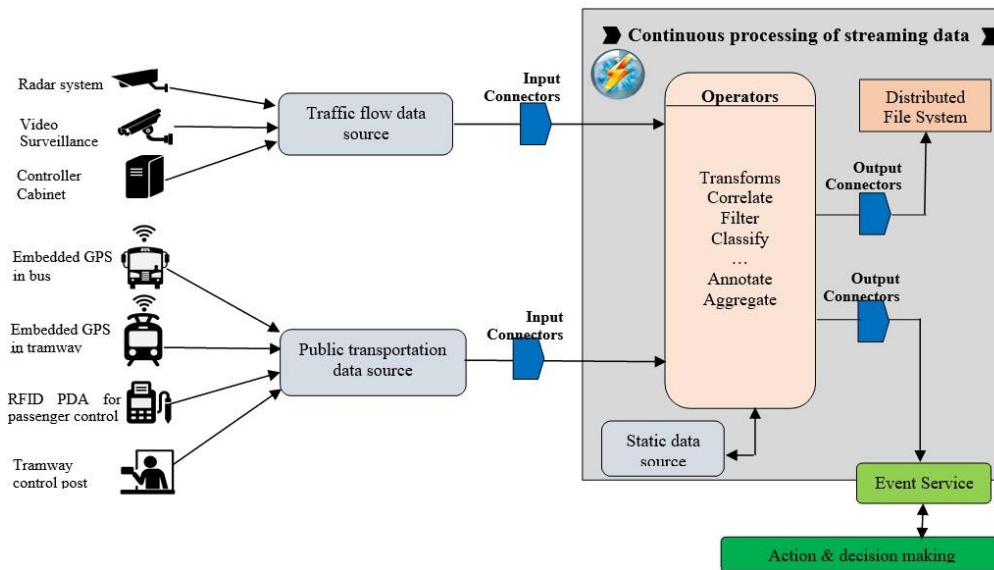


Fig. 2: Overall view of a traffic stream pipeline

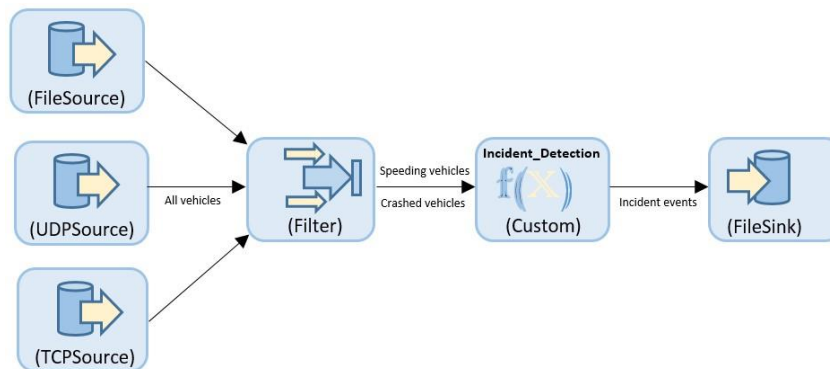


Fig. 3: Incident detection stream pipeline

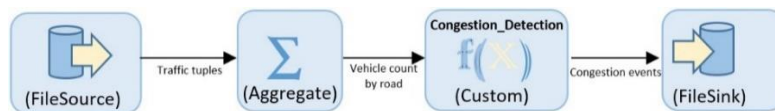


Fig. 4: Congestion detection stream pipeline

### Storage Layer

The traffic data flow gathered in real-time is stored in the storage layer and it includes road weather conditions, real-time weather information, vehicle tracking, passenger behavior, and associated occurrences. Traditional data storage methods and relational databases aren't designed to handle massive amounts of unstructured data and can't guarantee data scalability.

Hadoop framework Dean and Ghemawat (2008); Ghemawat *et al.* (2003) are used to manage enormous amounts of scalable and distributed traffic data. It allows our transportation system to handle petabytes of unstructured data distributed over thousands of nodes. Each node is equipped with a set of core processors and disks Arpaci-Dusseau *et al.* (1999); Ghemawat *et al.* (2003). The cluster is made up of a network of racks that are connected by a master/slave topology and interact using a Hadoop-specific block protocol Fox *et al.* (1997). Each rack contains multiple nodes, Fig. 5(a). The master node plays the role of Name Node and Job Tracker, it manages the HDFS and coordinates the execution of MapReduce programs on the slave nodes. The slave nodes play the role of Data Node and Task Tracker, they only store the datasets and execute MapReduce programs.

Our system is built to withstand hardware failures and is built around two major components:

1. The Hadoop Distributed File System (HDFS), which stores traffic data
2. The MapReduce engine, which processes data across the whole cluster

By duplicating data throughout the whole cluster, HDFS achieves dependability by replicating data across several nodes and assuming nodes may fail. Ghemawat *et al.* (2003); Liskov *et al.* (1991), Fig. 5(b). To avoid hardware failures, traffic data are split into block files of 128MB, and each file is stored on a DataNode. To prevent the system from failing, three replicas of each block file are configured by default. Hadoop transmits the MapReduce algorithm to each node to process the traffic data that it can access in the HDFS. This enables the cluster to process data more quickly and effectively than a traditional supercomputer design based on a distributed file system with data computations dispersed over high-speed networks using Gigabit Ethernet connections Howard *et al.* (1988).

### Experiments

The findings of experimental tests conducted in the city of Casablanca are presented in this section. This study was

made possible through a dataset produced by embedded systems in buses, tramways, and utility vehicles. The performance of our transportation system is evaluated on its ability to widely cover information from GPS sensors. The objective is to show that our system can instantly process data from 944 GPS access points (866 GPS access points for the buses and 78 GPS access points for the tramways).

### Dataset

The traffic data are varied, unstructured, and come in different formats. Two categories of data have been collected: Static and stream data. The OpenStreetMap data from Casablanca, public transportation lines (buses and tramways), GPS devices placed in public vehicles, historical vehicle tour schedules, passenger mobility, and real-time data from traffic control stations are used.

Bus data is centralized in four zones, with 70 lines spanning 1250 kilometers and many recordings of the behavior of 21000 children transported daily. The tramway's data includes a 31-kilometer fork-shaped line with 50 stations and several records detailing the daily behavior of 120 000 people. Streams data from road construction and maintenance activities, as well as planned events with a small or large audience and vehicle tour schedules (this activity is usually completed weekly), are used.

### Test Environment

For experimental tests, the stream application was deployed on a 5-nodes cluster consisting of 1 master and 4 slaves. Each node is equipped with a processor Intel (R) Core i5-2410M CPU 2.30GHz (4 CPUs) with 8GB RAM running on Red Hat Enterprise Linux6 64 OS.

### Urban Mobility Analysis

Every day, about 10 million transportation moves are made in Casablanca city. The average waiting time at peak hours is increasingly longer, Fig. 6. The road network configuration of the city is the major source of congestion in some districts, and the traffic utility vehicles may suffer a significant downturn. Figure 7(a) shows the daily traffic loads by district based on vehicle counting by using an inductive loop. Indeed, a significant flow of vehicles in the southwest, east, and downtown is witnessed. As a result, this causes main points of congestion and traffic jams. Figure 7(b) shows the average traffic speed in the district at peak hours. It has been remarked that areas with high mobility are more affected by downturns creating a long time of traffic congestion.

### Analysis of Number of Stream Tuples

All embedded systems communicate directly with our system via TCP/IP and UDP/IP protocols. Each embedded system sends information about vehicle locations, speed, stop, and entries-exit of passengers. Figure 8 shows the average number of GPS access points processed per second. Indeed, the process streams are distributed across all nodes and run simultaneously, this is called parallel computing. In our test, the distribution of stream process is automatically managed by the Stream Scheduler of Infosphere Streams. Each node can manage a set of stream data from multiple GPS access points. Each GPS access point corresponds to a tuple and it is managed by the source operators. Note also that the throughput processed by the system is influenced by node configurations (CPU and RAM).

### Analysis of Traffic Events

The system presents better performance, it consumes few resources of ram and every second, 2 million GPS access points are processed. The experimental values of frauds and incident detections from streaming data are shown in Table 1. All captured events in the table with the total number of frauds, number of detected frauds, total number of collisions, and number of detected collisions will be reported. So, the relative gap between frauds and captured frauds and the relative gap between collisions and captured collisions have been reported.

Based on the results provided in column 4, it is possible to deduce that the proposed fraud detection technique provides the best results and captures all fraud events (% fraud = 100%). In the case of collision events (column 7), It has been remarked that the quality of results decreases with the increasing number of vehicle streams. This is caused by a large number of vehicles with abnormal behavior which stop temporarily along the roadsides. In this case, it is

difficult to deduce if it is an accident or a temporary stop. But on a highway, it can be concluded that it is an accident or a breakdown.

### Performance Evaluation

In this section, we compare our system with the IBM InfoSphere Streams proposed by Biem *et al.* (2010a) for the city of Dublin. For the performance measurement, our comparison metric is based on the cluster nodes configuration and GPS tuples collected. As we raise the total number of nodes utilized for the entire stream application, the metric is the throughput, which is calculated as the number of tuples per second processed by the source operators. The experiment tests of Biem *et al.* (2010a) were carried out on a cluster of 64bit Intel XeonR quad-core machines running at 3Ghz with 16G bytes of memory.

Figure 9 shows the performance of the ITS application with an increasing number of nodes. The y-axis shows the throughput and the x-axis shows the total number of nodes used for the entire application.

Compared to the transport system proposed by Biem *et al.* (2010a), our ITS presents a good performance in terms of tuple collection.

For example, with 5 nodes, our ITS processes about 120 000 tuples per second against 80 000 tuples per second with the system of Biem *et al.* (2010a). Thus, this value increases with the addition of new nodes in the cluster. Overall, our ITS is 1.35 times more efficient than Biem *et al.* (2010a) system. Moreover, it is less greedy in terms of hardware resources. It works with a large number of GPS tuples with a cluster equipped with an Intel(R) Core i5-2410M CPU 2.30 GHz (4 CPUs) with 8GB RAM while the cluster nodes of Biem *et al.* (2010a) are equipped with an Intel XeonR quad-core machines running at 3 Ghz with 16G bytes of memory.

**Table 1:** Experimental results of event detections

Stream #Tuples	Fraud event			Collision even		
	Total	Detected	Ratio	Total	Detected	Ratio
$10 \times 10^3$	63	63	100%	12	12	100.00%
$50 \times 10^3$	96	96	100%	29	29	100.00%
$10 \times 10^4$	188	188	100%	34	34	100.00%
$50 \times 10^4$	277	277	100%	45	48	106.60%
$1 \times 10^6$	427	427	100%	53	58	109.40%
$5 \times 10^6$	524	524	100%	74	80	108.10%
$10 \times 10^6$	748	748	100%	86	92	106.90%
$25 \times 10^6$	1478	1478	100%	128	136	106.20%
$50 \times 10^6$	3472	3472	100%	222	253	113.90%
$75 \times 10^6$	4128	4128	100%	289	312	107.95%
$100 \times 10^6$	6174	6174	100%	417	448	107.43%

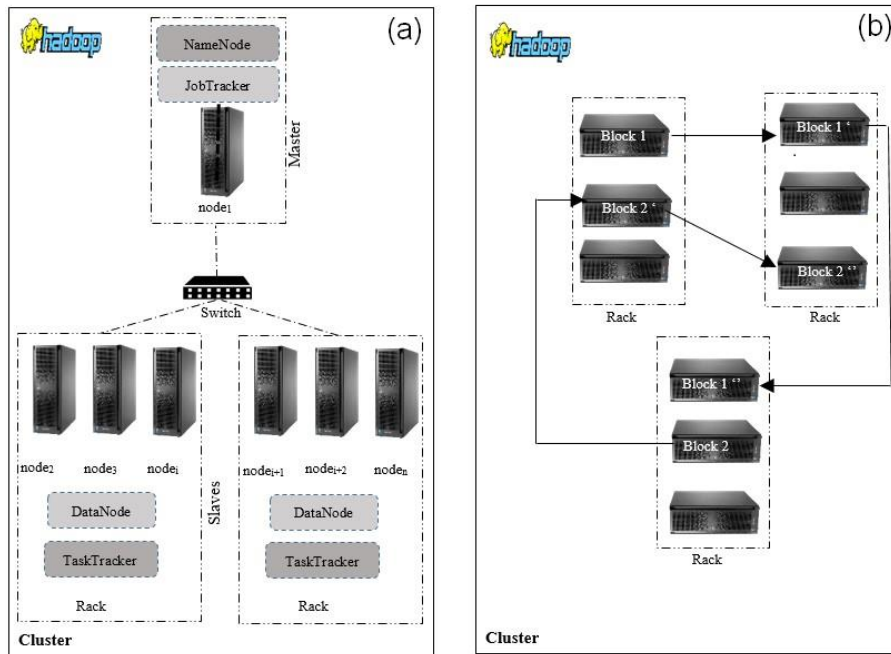


Fig. 5: Multi-nodes cluster of our transportation system

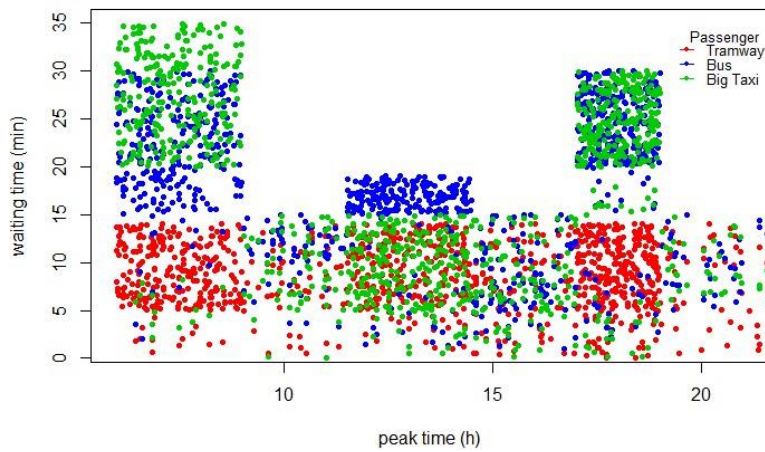


Fig. 6: Average passenger waiting time at peak hours

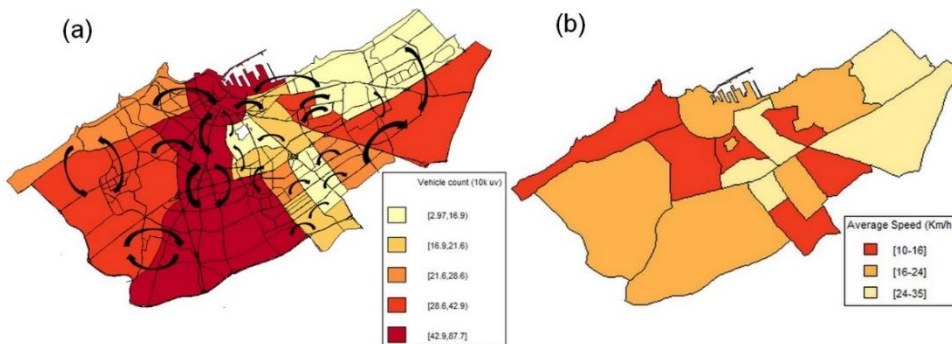
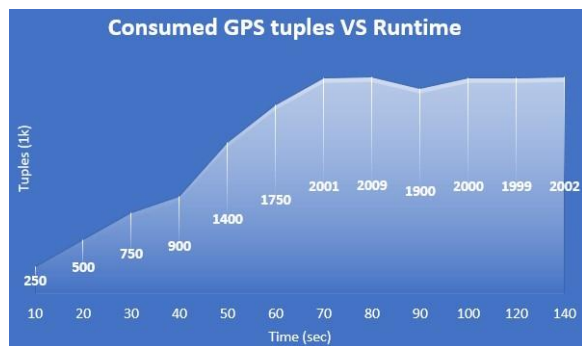
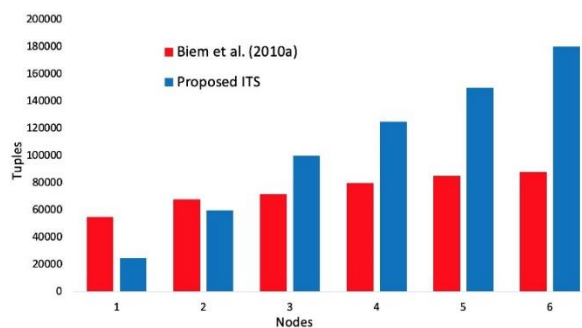


Fig. 7: Traffic flow of Casablanca city: (a) Daily traffic loads and mobility flow of different districts; (b) Average speed by the district at peak hours



**Fig. 8:** Number of GPS access points processed per second in the multi-nodes cluster



**Fig. 9:** Average number of tuples processed by second with the increasing cluster nodes

## Conclusion and Further Work

In this study, an innovative architecture for an intelligent transportation system based on the Hadoop big data framework is proposed. The system's ability to handle enormous amounts of traffic data while maintaining fault tolerance and providing real-time traffic flow monitoring has been experimentally proven. It was made possible by using Infosphere Big Insights and our system is well-equipped to meet scalability and adaptation difficulties. It has been also shown how the system can gather and manage traffic data from an increasing number of GPS access points with different transfer protocols. The streaming process has been used for real-time detection of frauds, accidents, and congestion events from the traffic flow. Our system satisfies the 4V criteria of big data and faces the challenges of storing large size and diverse variety of traffic data and supporting the scalability of traffic data. In terms of velocity and veracity, the response time and relevance of information made available to end-users are instant with a high quality of precision. For further work, we will: (1) Provide a path planning algorithm based on A\* search Bell (2009) and combine different variants such as users' preferences, weather, and traffic conditions; (2) analyze traffic flow and suggest efficient solutions

to urban transportation planning agencies, (3) Apply our proposed framework on other datasets and (4) propose a stream process application which can relate all vehicle incidents to emergency services.

## Author's Contributions

**Wilfried Yves Hamilton Adoni and Najib Ben Aoun:** Have developed the theory and performed the computations.

**Tarik Nahhal:** Has helped in the implementation part as well as the system design.

**Moez Krichen:** Conducted the literature review and performed the analysis.

**Mohammed Y. Alzahrani:** Has provided the first paper draft and interpreted the results and.

**Franck Kalala Mutombo:** Has supervised all the work and validated the idea. All the authors have discussed the results and approved the final version of the paper.

## Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

## References

- Abbas, A., Krichen, M., Alroobaea, R., Malebary, S., Tariq, U., & Jalil Piran, M. (2021). Opportunistic data dissemination for autonomous vehicles communication. *Soft Computing*, 25(18), 11899-11912. <https://doi.org/10.1007/s00500-020-05542-y>
- Adoni, W. Y. H., Nahhal, T., Aghezzaf, B., & Elbyed, A. (2017a, April). The MapReduce-based approach to improve vehicle controls during big traffic events. In *2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)* (pp. 1-6). IEEE. <https://ieeexplore.ieee.org/abstract/document/7962864>
- Adoni, W. Y. H., Nahhal, T., Aghezzaf, B., & Elbyed, A. (2017b, May). MRA\*: Parallel and distributed path in the large-scale graph using mapReduce-A\* based approach. In *International Symposium on Ubiquitous Networking* (pp. 390-401). Springer, Cham. [https://doi.org/10.1007/978-3-319-68179-5\\_34](https://doi.org/10.1007/978-3-319-68179-5_34)
- Adoni, W. Y. H., Nahhal, T., Aghezzaf, B., & Elbyed, A. (2018). The MapReduce-based approach to improve the shortest path computation in large-scale road networks: the case of A\* algorithm. *Journal of Big Data*, 5(1), 1-24. <https://doi.org/10.1186/s40537-018-0125-8>
- Aoun, N. B., Elghazel, H., & Amar, C. B. (2011, April). Graph modeling-based video event detection. In *2011 International Conference on Innovations in Information Technology* (pp. 114-117). IEEE. <https://ieeexplore.ieee.org/abstract/document/5893799>



- Aoun, N. B., Mejdoub, M., & Amar, C. B. (2014, May). Bag of sub-graphs for video event recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1547-1551). IEEE. <https://ieeexplore.ieee.org/abstract/document/6853857>
- Arpaci-Dusseau, R. H. anderson, E., Treuhaft, N., Culler, D. E., Hellerstein, J. M., Patterson, D., & Yelick, K. (1999, May). Cluster I/O with River: Making the fast case common. In *Proceedings of the sixth workshop on I/O in parallel and distributed systems* (pp. 10-22). <https://doi.org/10.1145/301816.301823>
- Bell, M. G. (2009). Hyperstar: A multi-path Astar algorithm for risk-averse vehicle navigation. *Transportation Research Part B: Methodological*, 43(1), 97-107. <https://doi.org/10.1016/j.trb.2008.05.010>
- Biem, A., Bouillet, E., Feng, H., Ranganathan, A., Riabov, A., Verscheure, O., ... & Moran, C. (2010a, June). Ibm infosphere streams for scalable, real-time, intelligent transportation services. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 1093-1104). <https://doi.org/10.1145/1807167.1807291>
- Biem, A., Bouillet, E., Feng, H., Ranganathan, A., Riabov, A., Verscheure, O., ... & Güç, B. (2010b). Real-Time Traffic Information Management using Stream Computing. *IEEE Data Eng. Bull.*, 33(2), 64-68.
- Bouillet, E., & Ranganathan, A. (2010, May). Scalable, Real-time Map-Matching using IBM's System S. In *2010 Eleventh International Conference on Mobile Data Management* (pp. 249-257). IEEE. <https://ieeexplore.ieee.org/abstract/document/5489658>
- Brahimi, S., Aoun, N. B., & Amar, C. B. (2017, March). Very deep recurrent convolutional neural network for object recognition. In *Ninth International Conference on Machine Vision (Icmv 2016)* (Vol. 10341, pp. 28-32). SPIE. <https://doi.org/10.1117/12.2268672>
- Brahimi, S., Aoun, N. B., & Amar, C. B. (2018a, October). Improved very deep recurrent convolutional neural network for object recognition. In *2018 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (pp. 2497-2502). IEEE. <https://ieeexplore.ieee.org/abstract/document/8616424>
- Brahimi, S., Ben Aoun, N., Ben Amar, C., Benoit, A., & Lambert, P. (2018b). Multiscale fully convolutional densenet for semantic segmentation. <https://otik.uk.zcu.cz/handle/11025/34596>
- Brahimi, S., Ben Aoun, N., Benoit, A., Lambert, P., & Ben Amar, C. (2019). Semantic segmentation using reinforced fully convolutional densenet with the multiscale kernel. *Multimedia Tools and Applications*, 78(15), 22077-22098. <https://doi.org/10.1007/s11042-019-7430-x>
- Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107-113. <https://doi.org/10.1145/1327452.1327492>
- Dodge, M., & Kitchin, R. (2007). The automatic management of drivers and driving spaces. *Geoforum*, 38(2), 264-275. <https://doi.org/10.1016/j.geoforum.2006.08.004>
- Dong, H., Wu, M., Ding, X., Chu, L., Jia, L., Qin, Y., & Zhou, X. (2015). Traffic zone division based on big data from mobile phone base stations. *Transportation Research Part C: Emerging Technologies*, 58, 278-291. <https://doi.org/10.1016/j.trc.2015.06.007>
- Fox, A., Gribble, S. D., Chawathe, Y., Brewer, E. A., & Gauthier, P. (1997, October). Cluster-based scalable network services. In *Proceedings of the sixteenth ACM symposium on Operating systems principles* (pp. 78-91). <https://doi.org/10.1145/268998.266662>
- Ghemawat, S., Gobioff, H., & Leung, S. T. (2003, October). The Google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles* (pp. 29-43). <https://dl.acm.org/doi/pdf/10.1145/945445.945450>
- Hao, J., Zhu, J., & Zhong, R. (2015). The rise of big data on urban studies and planning practices in China: Review and open research issues. *Journal of Urban Management*, 4(2), 92-124. <https://doi.org/10.1016/j.jjum.2015.11.002>
- Howard, J. H., Kazar, M. L., Menees, S. G., Nichols, D. A., Satyanarayanan, M., Sidebotham, R. N., & West, M. J. (1988). Scale and performance in a distributed file system. *ACM Transactions on Computer Systems (TOCS)*, 6(1), 51-81. <https://dl.acm.org/doi/abs/10.1145/35037.35059>
- Jabbar, R., Fetais, N., Kharbeche, M., Krichen, M., Barkaoui, K., & Shinoy, M. (2021). Blockchain for the Internet of vehicles: How to use blockchain to secure vehicle-to-everything (V2X) communication and payment? *IEEE Sensors Journal*, 21(14), 15807-15823. <https://ieeexplore.ieee.org/abstract/document/9363191>
- Jabbar, R., Kharbeche, M., Al-Khalifa, K., Krichen, M., & Barkaoui, K. (2020). Blockchain for the internet of vehicles: A decentralized IoT solution for vehicle communication using Ethereum. *Sensors*, 20(14), 3928. <https://www.mdpi.com/1424-8220/20/14/3928>
- Kitchin, R. (2014). The real-time city? Big data and smart urbanism. *GeoJournal*, 79(1), 1-14. <https://link.springer.com/article/10.1007/S10708-013-9516-8>
- Krichen, M. (2021). Anomalies detection through smartphone sensors: A review. *IEEE Sensors Journal*, 21(6), 7207-7217. <https://ieeexplore.ieee.org/abstract/document/9327468>
- Lécué, F., Tallewi-Diotallewi, S., Hayes, J., Tucker, R., Bicer, V., Sbodio, M., & Tommasi, P. (2014). Smart traffic analytics in the semantic web with STAR-CITY: Scenarios, system, and lessons learned in Dublin City. *Journal of Web Semantics*, 27, 26-33. <https://doi.org/10.1016/j.websem.2014.07.002>

- Li, L., Su, X., Wang, Y., Lin, Y., Li, Z., & Li, Y. (2015). Robust causal dependence mining in the big data network and its application to traffic flow predictions. *Transportation Research Part C: Emerging Technologies*, 58, 292-307.  
<https://doi.org/10.1016/j.trc.2015.03.003>
- Liskov, B., Ghemawat, S., Gruber, R., Johnson, P., Shriram, L., & Williams, M. (1991). Replication in the Harp file system. *ACM SIGOPS Operating Systems Review*, 25(5), 226-238.  
<https://doi.org/10.1145/121133.121169>
- Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute.
- Toole, J. L., Colak, S., Sturt, B., Alexander, L. P., Evsukoff, A., & González, M. C. (2015). The path most traveled: Travel demand estimation using big data resources. *Transportation Research Part C: Emerging Technologies*, 58, 162-177.  
<https://doi.org/10.1016/j.trc.2015.04.022>
- Zheng, Y., Liu, F., & Hsieh, H. P. (2013, August). U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1436-1444).  
<https://dl.acm.org/doi/abs/10.1145/2487575.2488188>