Article

Iot Powered by Big Data: Architecture, Ecosystem, Applications

Ahmad Alflahat *

Posted Date: 5 September 2023

doi: 10.20944/preprints202309.0148.v1

Keywords: Internet of Things; Big Data Ecosystem; Hadoop Ecosystem; Storage Computing

Article

IoT Powered by Big Data: Architecture, Ecosystem, Applications

Ahmad Alflahat [1,†,‡]

[1] Affiliation 1; ahmadtariqf@gmail.com
* Correspondence: ahmadtariqf@gmail.com;
† Current address: Petra, Maan, Jordan

Abstract: To handle the huge amount of data generated by IoT devices, Big Data processing tools make it easier. This paper discusses the Big Data concept and its main V's characteristics. It further describes IoT-enabling technologies; nominally cloud computing such as SaaS and PaaS. The centralization and infrastructure of Big Data systems, and how Cloud Computing gives a platform access to the data from anywhere. The paper explores IoT with big data architectural solutions for various use cases across the healthcare and transportation sectors.

Keywords: Internet of Things; big data ecosystem; hadoop ecosystem; storage computing

1. Introduction

The term of Big Data has become the orientation of the industrial and researcher, next year data will grow immensely compared to previous one. Big Data contain process techniques and analytics tools, to check the level of insight. Descriptive by processing the data of past and know why it has happened. Predictive make a prediction of how things are going to be in the next time, things like data. Prescriptive how to deal with the result and get the benefit of it to the company. Building Big Data architecture, and get the optimal for what use case you have, has become a hot work job called a data scientist. Scientists are professional in using Big Data ecosystem tools; they are able to build an optimal solution for the scenario of the use case. Eco-system tools are developed by many organizations care about the Big Data and the good thing is most of them are open source, so free of charge, which is good for the provider company of the platform. IoT's notion is to require a good vary of things and switch them into good objects — something from vehicles, watches, fridges, and tracks on the railway. Computer chips and sensors square measure sometimes designed for the aim of grouping knowledge for things that would not be connected to the net and capable of process and handling knowledge. Nonetheless, compared to the chips utilized in mobile devices, laptops and PCs, these chips are principally want to collect knowledge that defines product potency and usage patterns for customers. By new technology data increased and open up new approaches to handle that data by Big Data. Internet of Things (IoT) is one of the most famous challenges in this century. Where the two work along is in period or near-real-time cases of however you create pc information accessible to analytics tools. Take a state of affairs like prophetic maintenance wherever rules engines, stream analysis, machine learning, and method controls area unit accustomed to assist you find out about the device's output, results, and state, and mechanically react to programmatic or human interaction. Developing a technique to show information into unjust is essential to the success of massive information and IoT. These area units open-ended approaches that area unit adequate data quality with a rise within the variety of connected devices, the enterprise can have additional chances to use these tools to gather vital, helpful info that's improved in business opportunities. In this paper is organized as follows. Section II study big data characteristics and ecosystem tools, and show build up architecture by other researchers. Section III present the IoT layer and show that data processing layer is the umbrella for the big data system. In Section IV discuss cloud-computing services and mentioned some work on them. Use cases and scenario by many authors have been discussed in section V. Finally, section VI the conclusion and future work.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 September 2023 doi:10.20944/preprints202309.0148.v1

2. Big Data and Hadoop Architectural

This section describes the acquisition and how the process of gathering and filtering data after it is generated from the IoT-Sensors before it gets into the warehouse or any other storage solution. Data is acquired by the rules of the characteristic of big data volume, variety, and velocity.

2.1. Big Data Characteristics

Volume: The most important future for big data means the large scale of data that the systems deal with [**?**]. Salah et al. in their study on data stream curation, suggest that data generated from IoT devices would require reduction of volume. They state that issues that can impact process automation to curate data in order to apply high-quality data-driven learning models on data consumed in large data systems are complex, such as structured and unstructured physical IoT devices with high volume characteristics, fast data rate and unreliable performance that would require reducing data volume.

Variety: Heterogeneity, data do not have a stander of how they look on while storing it [**?**]). The different of source data come from creating the different of data structure, like in IoT GPS, RFID, temperature sensors, etc., each one has its semantics of sending the data to store, like analog, digital, etc. [**?**].

Velocity: the measure of how quick the data is coming in, ingestion and retrieve, refer to the type of speed like the batch, stream and online processing. In addition, the change in the rate of that speed is considered [**?**].

2.2. Common Use Ecosystem Tools

Widely known framework help with analysis data using parallel processing of large data sets. The architecture of Hadoop based on distributed machines, scaled from single to thousands of machines, each machine (cluster) handles collect, process, store, and much more. This section will discuss the most commonly used tools in the Big Data system.

1. Hadoop distributed file system (HDFS)
   Storage system it's schema designed to be distributed and resilient, designed to let the use of MapReduce easy, but need other components (like Spark, BI tools, and MongoDB, etc.) need Apache YARN to make the interaction with the HDFS easily [**?**]. HDFS provides aggregation increasing bandwidth by bonding the cluster to boost the throughout [**?**]. Master/Slave is the design of HDFS, the master called the NameNode and the Slave is the DataNode, each cluster has one NameNode and more than two DataNode. Data split in HDFS for several blocks and each block replicated multiple times each block has a default size of (128/64MB), then each block of the same data replicas stored in a different node [**?**]. NameNode is responsible for making the decision of how the replication for the blocks and where each block is stored. NameNode holds the metadata of all the data in the same cluster. HDFS allows you to write once but read many [**?**].
   The reason for HDFS is write-once-read-many when Google Developed HDFS the only purpose is to receive data for once, and do the process on them many times, in other words, the batch process. For that, using HBase on top of HDFS is necessary when you want to be able to write and read randomly in the HDFS. HBase is a non-rotational database run on top of HDFS, let random/querying capabilities. Also, it stores the data as key-value pairs unlike HDFS it stores data as a flat-file. HBase can help other resources through the shell command in like Java, Avro or Thrift.
2. MapReduce Open source under the developed by Google [**?**], based on java, MapReduce model is based on two phases/tasks, the map task is dividing the data into several blocks, the result of dividing will be two blocks one of them will be the input key of the data and the other block will be the data itself and is called the value, in the end, the document will contain all the (key, value) in it for all the data [**?**]. MapReduce's main task is to allow the system compatible with the large

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 September 2023 doi:10.20944/preprints202309.0148.v1

sets of data, especially in a distributed manner [?]. The ability of MapReduce is to process data and the fact that the query of execution is shorter compared with SQL/RDBMS.
Satoh, I. [?] propose an architecture for MapReduce in IoT applications, it's slightly different from the original one the mapper and reducer node, where there are three nodes the mapper and reducer, also adding the worker node. Figure ?? shows the integration between the three nodes. Each connected in its way and every connection has its own capability. Mapper and Worker are the response for the duplications and deployment of tasks. Worker alone is the response for three function application data processing like reading, process and store data, each one is standalone can be called without the other functions. The last one, Worker and Reducer for reducing data process results the is the response on how the data stored. According to Abdallat et al. [?], Hadoop MapReduce can be viewed as a dynamic ecosystem to be studied for the job scheduling algorithms to draw a clear image.

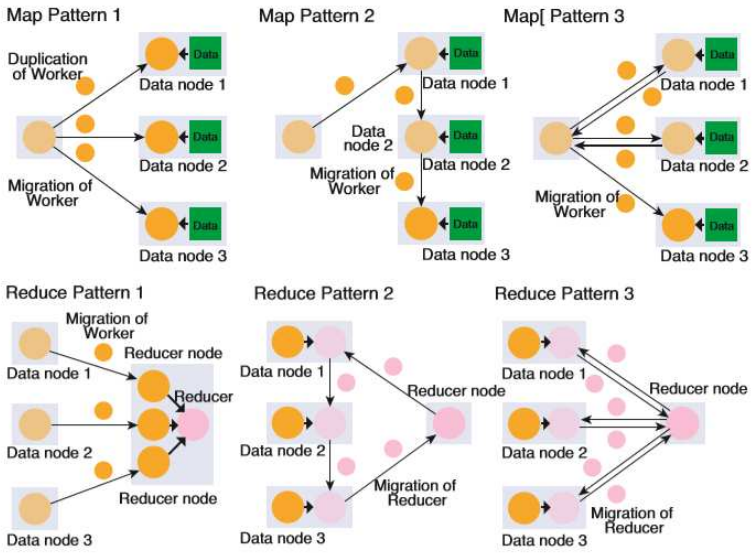

Figure 1. MapReduce proposed by Satoh, I.

3. YARN Stands for Yet Another Resource Negotiator, allows the different ecosystem to be integrated with HDFS, the task of YARN is resource management and scheduling in Hadoop distributed processing frameworks [?]. Used for allowing the system to be able to deal like storing and processing data by newer process methods like stream process, graph process.
4. Spark Ounacer et al. said it is open-source under the license of Apache Foundation, spark includes multiple tools for different propose, it's used in distributed computing. Spark is way better than MapReduce, the spark is faster in both batch and in-memory processing/analysis. Spark contains not only the MapReduce look-alike function but also it has other tools integrated to it, they are: Spark SQL, Spark Streaming, MLib, GraphX and SparkR, all these tools are built on top of the Apache Spark Core API.
   The process of Big Data in a real-time or stream online, Spark integrated with Kafka is part of the architecture of the system [?]. Spark can run in 3 cluster mode types the standalone (native spark structure), Hadoop YARN, or Mesos. Also, Spark support distributed storage includes HDFS, Cassandra, Amazon S3 [?]. Spark Streaming is part of the Apache Spark that responsible for handling computing the different data which is coming to fast to the system [?], it allows doing several functions on the streamed data (like obtain data, joining the stream, filtering stream, etc.) [?].
5. Hive Used to handle the reading and writing, moreover the managing of large data using SQL like interface. Hive is an SQL engine that integrated with the Hadoop HDFS and runs MapReduce jobs, most likely hive use for the analytical queries [?].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 September 2023 doi:10.20944/preprints202309.0148.v1

2.3. Ecosystem

There are many tools in the big data ecosystem each one has it is own unique functionality when building architecture for the system we integrate tools with each other to have the expected result. This section will discuss some architecture proposed by other authors.
Jesse, N. [**?**] proposes the Hadoop ecosystem for value chain (smart factories) based on the Lambda architecture to solve the latency problem. Include HDFS for the storage, for the distributed processing by using MapReduce, scheduling and cluster resource YARN, Sqoop for the SQL commands by importing and exporting. Based on Apache Kafka for the ingestion to deal with streaming data, Hive and Impala for querying and summarization, Mahout for the Machine Learning, and Pig for the data-flow.

Ta-Shma, P. et al. [**?**] design Hut architecture for smart cities in IoT. The suggested components, for file storage, open-source called OpenStack Swift been mentioned, and for the data acquisition, NodeRed is suggested due to the ability to deal with the different data types (XML, JSON, etc.). Data ingestion Secor integrated with the message broker Kafka, so the system can be able to handle big backlogs by Kafka coming from the Secor. In addition, Kafka supports batch and online streaming. Elastic Search based in Solr or Lucene for searching and index. For the analytics Spark the ability to integrate with many storage systems (S3, Hadoop, etc.) and the RDD to process in the memory without the need for replications. However, the proposed solution contains some of the Spark library Spark SQL and Spark MLlib, because of the schema of Spark SQL "DataFrame" used for the advantage of Elastic Search to search for metadata. MLlib has popular machine learning algorithms and the algorithms of classification, clustering, and filtering and dimensional reduction.

Chou, S. et al. [**?**] study the data accessing for electric loads warehouse. Due to the use of historical data, they needed a tool to handle by using Spark SQL for scheduling module, and Apache Sqoop for the ingestion for storing the data from SQL form to Hive data type format. Hive is a tool for defining the table schema not for storing so they use Hive on top of HDFS as file storage and MapReduce for the parallel distributed. Taking advantage of the existing of Hive, they integrated it with the Cloudera Impala for the Search engine by sharing the tables and the data between both components, they use Impala for the low latency and high performance in querying the data that are stored in HBase and Hive. Selecting Hue for the interface and visualization, by using this tool monitoring the process would be easier also for the dynamic dashboard search by using the Solr.

After implementing the architecture the results show, that creating a table with Impala is faster than creating it with Hive, the difference explained due to the process that has occurred to the data, Impala based on caches node to be used in future work and the process starts at the boot time. However, Spark shows a much better way of execution time for ETL, the reason for that is the way of Hadoop deal with data by splitting them into three copies and which mean that it splits the pipeline mission into more nodes, which means the more slave node the more latency. For statistical processing, it shows that Spark takes less time than Hive. In the end, they show the difference between Spark and Hive in the Write/Read process as shown in Figure **??**, Spark is much better in both reading and writing.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 September 2023 doi:10.20944/preprints202309.0148.v1

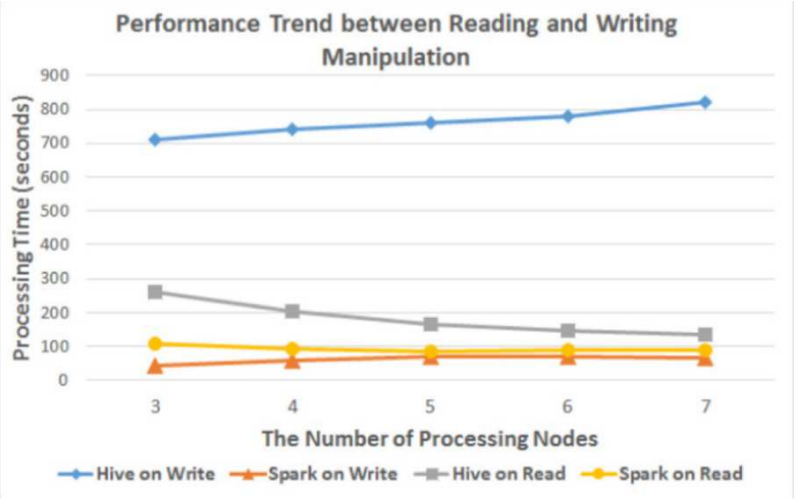

Figure 2. Time respond for write/read in Spark and Hive.

3. IoT Architectural

Internet of Things system level features have devices, data exchange , usage of wireless technologies, scalability of data, semantic interoperability, as well as privacy and security measures [**?**]. The internet of things (IoT) includes cloud, things and application interfaces [**?**]. These interfaces govern the communication between entities, such as humans or objects, and IoT service architecture. The IoT architecture is divided into four main parts which communicate with each other. These four main layers are detailed as:

- Data perception layer:
  It has the ability to perceive, detect, and collect information and objects, and collect data. using different devices such as sensors and RFID [**?**]. The data come from different sources such as sensors, RFID as induction loop detectors, microwave radars, video surveillance, remote sensing, radio frequency identification data, and GPS [**?**]. This layer handles data through different IoT sources and afterward totals that data. In some cases data have security, protection, and quality prerequisites. Likewise, in sensor information, the metadata is constantly more noteworthy than the genuine measure. So, filtration methods are connected in this layer, which spot the pointless metadata [**?**]. According to Abuqabita et al., the main challenge facing big data is IOT information acquisition. The IOT device's need for infrastructure produces ongoing data streams and develops methods to derive good information from these data, analyzing them through machine learning and artificial intelligence approach is the only way to work with IOT prospective big data.
- Network communication layer:
  This layer facilitates the information between sensors [**?**]. It transfers collected data from the perception layer to the following layer using communication and internet technologies [**?**]. This layer is divided into two sub-layers which are the data exchange sub-layer that handles the transfer of data and information integration sub-layer which aggregates, cleans and fuses collected data [**?**].
- Data processing layer:
  The bulk of IoT architecture is in this layer. It is in charge of preparing the information [**?**]. In a big data concept, information is processed in parallel using Hadoop distributed file system. Apache Spark with its different components is commonly used as well. For further information about this layer, the reader is referred to section IV.
- Application layer:
  It aims to create a smart environment. Hence, it receives the information and process content to deliver intelligent services to different users [**?**]. It may also include a support layer. The final

 doi:10.20944/preprints202309.0148.v1

decision is often made in this layer. This layer commonly uses interface oriented setting which enables the entire IoT system to perform under high efficiency.

4. Cloud Services

Collins, E. [**?**] when it comes to giving a relation between big data and cloud computing, cloud service and each level that it provide. Business looking for an application like SaaS to take care of the analysis, collecting and store the data, and the most important one is the visualization to the end-user. Other businesses looking to reduce the cost, at the same time they have control over some of their resources by using IaaS or PaaS, also it makes the interaction for the non-developer easier. The relation between big data and cloud computing enabled converged analytics, in another word, many different resources can do the analytics on the data. There are three common types of cloud service:

1. Software as a Service (SaaS)
   Gives a utility of managing from the Clint's side, and the remote server, also its on-demand applications which means it can be accessible from the internet.
2. Platform as a Service (PaaS)
   Provided more flexible system, developers can build there customized system, and they don't need to have any thinking about the like where to store the data, or how is the connection between the components in the system and the maintaining, all of this is been take care of by the provider of the service.
3. Infrastructure as a Service (IaaS)
   Its more scalable than the other services, provides the authorization to develop and make the change on the storage and monitor the computers in the network, and other services, in other words, it's on-demand resources.

Elshawi, R. et al. [**?**] study Big Data as a services, considering two of the cloud services are integrated with big data, the PaaS contain the tools are responsible for the analytics and the implementation of data science, and SaaS contain the tools are responsible for extracting knowledge from the data. Mention the most used tools in each layer, like MapReduce in PaaS and other like OpenStack, HDInsight, Apache Spark, Sector/Sphere, AzureML, etc. SaaS in the other hand is a way to simplified to the end user to analysis the data and develop the process. How High Level Language (HLL) is part of SaaS, which is a programming language mostly used R System due to the ease-of-use and the wide community. Toolkits and libraries is part of the SaaS layer, well known one is the Apache Mahout because the particle and scalable for the machine learning algorithms especially for the distributed systems. Also the mentioned the Declarative Interfaces/Languages and Cloud Machine Learning Services and how they are part of the SaaS layer, and discuss the most popular tools and techniques for them.

5. Big Data Frameworks and Applications on IoT Domains

Powering IoT with big data could provide very helpful and fruitful solutions which would increase efficiency and produce new frameworks which facilitate different customized solutions. IoT powered by Big Data is found in multiple applications. In this survey we consider applications in the Healthcare and Transportation sectors, as recent development in those two sectors is excessive and rather impressive.

5.1. Healthcare

At the recent time, there has been an increased use of IoT in healthcare provision, a phenomenon which has been recognised as a revolution in Healthcare of the 21st century. One of the proposed systems for processing is the Hadoop-based System. Big Data and IoT could be used in the healthcare sector in a wide range of applications such as monitoring, prevention, and evaluation of medical

conditions.
Rathore, Ahmad, and Paul [**?**], propose a medical emergency management system based on Hadoop ecosystem using IoT technology. Researchers argues that the increased use of Internet of things in the health sector results in the generation of massive data that requires management. They indicate that the management of the enormous data is hard, considering most actions in the medical sector require a sound system. In their proposal, they advocate for the Hadoop Ecosystem for addressing emergency conditions in the medical industry. The system uses IoT technology to manage this large volume of data. They define the Internet of Things as the system of interrelated computer devices and digital devices, equipment, animals, or individuals that are given unique identifiers and have the ability to share data over a network independently without the use of computers or human interaction [**?**].
The Hadoop system would be suitable for managing the increased data generation in medical health. The massive data generation attributes to the deployment of sensors in managing health conditions in patients. Various sensors are embedded in our bodies to track activities, our health status, and others to coordinate our activities. The sensors are attached to multiple parts of our organs such as heart, wrists, ankles, and cyclist helmets and gather a lot of data used in the medical industry. It can aid in assessing diabetes, blood pressure, body temperature, and other data types. The data is coordinated at a central point where the sensors send data [**?**]. The architecture of the Hadoop Ecosystem consists of the following components:

- Intelligent building- it houses the collection, processing unit, aggregation result unit, and the application layer services. It is the primary part that incorporates the intelligent system to capture big high-speed data from the sensors. It receives processes and analyses medical data from the multiple sensors attached to the patient body.
- Collection point- this is the avenue where the collected data enters the system. It gathers data from all the registered individuals of the BAN network. It is served by one server that collects and filters data from the people.
- Hadoop processing unit- it consists of the master nodes and multiple data nodes. It stores data in blocks. It has data mappers that assess the data to determine its validity, whether normal or abnormal. Normal data requires analysis, while necessary data is discarded.
- Decision servers- they are built with an intelligent medical expert system, machine learning classifiers, together with some complex medical issue detection algorithms for detailed evaluation and making choices. Their function is to analyse the information from the processing unit.

The Hadoop ecosystem system implementation is done through the use of a single node application, UBUNTU. The evaluation of the system focuses on the average time taken in processing a given record from the various datasets.

Taher, Mallat, Agoulmine and El-Mawass [**?**] further suggested the use of an IoT-Cloud as the solution for the significant data challenges in health care. The system focuses on processing stream and batch data. The authors' system is based on Amazon Web Services tools which give us sufficient tools that can be used in the data processing. The system is cost-effective because it does not require the installation of other software to enhance its functionality. It has an option to set and run prototypes that excludes the costs of procuring other software and maintenance of infrastructure. The tools from the AWS adopted in the authors' system allows the user to capture data stream, compute, process, store, and give notifications. These tools include the following [**?**]:

- Amazon Kinesis stream – captures and stores terabytes of data per hour. The tool provides a 24-hour access of data from the time it is included in a stream.
- Amazon Elastic Compute Cloud (EC2) offers the user the chance to launch the servers of choice.
- AWS IoT core- the ore helps in connecting to the IoT devices, receiving data through the MQTT, and publishing the message to a given topic.
- Elastic MapReduce (EMR) - allows quick processing and storage of big data using the Apache Hadoop application.

The proposed system is applied in the healthcare industry to monitor patients' heart rhythm challenges. It is adopted in an electrocardiogram (ECG) where the electrodes are attached to the patient's chest to record the patient's heart electrical signal that makes the heartbeat. The ECG –view dataset operates on the proposition that to achieve better results from the analysis of big data, the data should be normal, collected within a given period and accompanied by a better-designed hypothesis made by a group of specialists. The data sets exist on a large scale and thus not suitable in processing healthcare information due to its confidentiality and sensitivity. However, the system depends on a surrogate dataset from other domains.

Chhowa, Rahman, Paul, and Ahmmed [**?**], in their study on IoT, generated big data in the health care setting. The authors argue that the prevailing challenges posed by big data can be monitored using deep learning algorithms. In their monitoring and analysis, the authors use a modern machine learning-based learning algorithms and proposals made by others to improve the speed of extracting files and accuracy. The system aims at providing better and reliable systems that doctors can use in monitoring critical patients. The article centres on providing individuals with a sustainable online health monitoring system that can access big data. The authors proposed the system uses supervised and unsupervised learning paradigms obtained from the ANN. They advocate for the DL algorithms since they offer self-extraction abilities. The system works like the human brain and requires neurons to process the necessary data[**?**].
The MD algorithm has three layers which are input, hidden layers, and the output layers. The architecture of the system is as follows:

- Conventional Neural Network – extracts vision-based characteristics through the use of hidden layers. The hidden layers in the CNN help in establishing chronic diseases in living things connected to the IoT.
- Long short term memory (LSTM) - controls the accessibility to memory cells and prevents the breakdown in connection arising from additional inputs. It works on three neuron gates, namely forget, read and write neurons.
- Restricted Boltzmann Machine (RBM) - comprises of both hidden and input layer. Aims at maximising the probability results of the visible layer.
- Deep Belief Network (DBNs) - they comprise of both the visible and hidden layers. They help in reconstructing and disarticulating training data that is presented hierarchically.
- Deep Reinforcement Learning (DRL) - focuses on enabling software agents to self-learn to generate useful policies to provide long term satisfaction of the medical needs. The problem associated with this system is that it does not have moderate data sets and cannot detect signals in offline mode.

Zameer, Saqib and Naidu [**?**] implement an IoT architecture with results showing considerable successes and benefits. The type of architecture described in the paper has three different layers, each one having a specific function to perform. These include the layer of data acquisition, storage and analysis, and finally, a layer of visualization and decision making support. It has been illustrated that it is possible to drastically reduce the rates of chronic illnesses as well as accidents in the event a strategic and systematic approach is employed when an IoT and Big Data is implemented in healthcare.

Rei, Brito and Sousa [**?**] made advancement to the field of big data in the health sector by executing a research to assess the IoT platform for collecting data and medical sensors. The authors propose a system that depends on IoT to store and monitor sensors' information in the healthcare setting. Their basis of evaluation centres on the Kaa IoT and Apache HBase. The architecture of the system points that the data originates from the patient. It has four parts the patient, IoT platform, data storage and the integrated analytic system. The Apache Cassandra and Apache HBase are adopted in the medical centres to process unstructured data. The Kaa IoT platform allows the user to store large volumes of data. The authors argue that the implementation of the system can be done in the Local area network (LAN). The results of the study state that the system provides real-time processing

techniques that allow evaluation and analysis of the health care sensor data. They assert that the review of the system is based on its capabilities to collect, write, and convey medical data [**?**].

Mezghani, Expoxito, and Drira [**?**] also had their input on the issue of big data in the health setting, arguing for a model-driven methodology. Their system focuses on a combined and instantiated mix of patterns that help in creating a flexible cognitive monitoring system. The system is used to monitor patient health obtained from multiple sensors. The system has two main phases, namely requirement identification and formalization. Requirement identification focuses on deliberation with the domain specialists to extract the system functionality capabilities and identifying the non-functional needs. The formalization phase concentrates on making the obtained data formal and structuring the highlighted requirements into a detailed and sound models that help in explaining how the system processes interactions. The article points out various pattern names, their context of use, problems, and their proposed uses. The phases are as follows:

1. Management Processes' coordination:
   Here the report identifies the blackboard pattern to be the reference point for coordinating management processes. This function is achieved through the other four patterns which are;
   - Knowledge pattern that focuses on the implementation of automatic computing. It solves the challenges of scalability and limitations during the centralization if knowledge in healthcare. The challenge can be eliminated through the decomposition of the knowledge for the better management of the IoT system into three categories. The sensory, context and procedural knowledge
   - Cognitive monitoring management pattern is designed for specific devices that generate data in one unit through the use of syntax and representation. It focuses on solving the problem of the new IoT devices that require new software to be created to necessitate its expensive functionality. The cognitive monitoring management pattern makes it easier by developing a system that uses human visualization to retrieve and receive notifications for any changes in the situation. Human visualization of the system allows the management of the system through setting modification and allowing the IoT system to get help from the specialists and acquire knowledge.
   - Predictive Cognitive Management pattern – it is an extension of the above pattern where it focuses on modeling and coordinating the monitoring, analysis and expert trends. It dissociates the interaction of the overseeing process with the sensory and setting data to provide new insights about the elements under examination.
2. Semantic integration:
   Semantic integration is another category of the system pattern that has the following subcategory:
   - Semantic knowledge mediator is used in a setting to enhance better management of all systems with IoT technology. It is based on the integration of several sources to obtain a detailed and better understanding of the business, setting, system and the environmental knowledge. The author stresses that the system would eliminate the challenges of heterogeneity associated with distributing and representing knowledge. The system can solve this problem by providing resources that foster the collaboration of various types of providers when analyzing the knowledge. It eliminates heterogeneity by providing flexible and extensive devices to contain new sources of know-how.
3. Big data and Scalability:
   Focuses on monitoring and assessing the primary processes facing big data problems. This is done by the big data stream detection pattern that focuses on reducing the velocity and volume of data to ensure its integration for real-time integration. This pattern solves these problems effectively and at low costs as opposed to the manual process that is time-consuming and costly. The Big data analytic predictive pattern also enhances the scalability. It is an extension of the above pattern. It focuses on supporting big data to manage batch processing. It borrows heavily

 doi:10.20944/preprints202309.0148.v1

from the human mind and how it processes information. It operates by importing data from external databases and harmonizes it in the long-term memory. The authors also argue for the management process multi-tenant pattern that instills scalability in data through dissociating and deployment of the primary functions[**?**]. Ma'arif, Setiawan, Priyanto and Cahyo [**?**] are other contributors in the issue of big data in the health care setting. They focus on developing a cost-effective machine that would help health centres to monitor and analyse sensor data effectively. The authors advocate for a system with the following architecture.

- Sensing unit – this unit collects vital information from the patient's body. It is constructed using the ESP2866 controlled by the NodeMCU. The NodeMCU gathers data not limited to heart rate, body temperature, and blood pressure.
- Data processing unit – this unit is developed according to the lambda architecture. Lambda architecture refers to a framework that is used to design data in a manner that allows the system to handle a substantial piece of data. The architecture combines the batch processing and fault-tolerance aspects that allow the system to secure latency and ensure real-time access to data respectively.
- The lambda architecture depends on three layers that increase its efficiency in handling massive data on time. The layers are speed, batch and serving. The speed layer focuses on the immediate data, assures a real-time view of the patient's important sign indicators as registered by the sensors. The batch layer manages historical data while the serving layer that indexes the batch view.
- Presentation unit – this unit contains mobile and web applications that help the user to access the data in the system. Our users here are health care providers.
- Communication protocol – this element has various parts specialised to offer coordination between the components of the system. For instance, the interface mediates the interaction between sensing and data processing unit; Application Programming interface connects processing unit and the presentation unit to ensure the user can access the information which is the result of the whole process.

5.2. Transportation

IoT applications powered by Big data, in the transportation sector mainly focus on harnessing state-of-the-art architectures to predict congestion and to facilitate the flow of cars in smart city. The applications considered also provide a good insight on how to integrate IoT technology with a smart car to provide an easier and more enjoyable driver experience.

Nkenyereye and Jang [**?**] proposed the use of querying model and obtaining useful information from CAN bus data through big data technology. This model deals with collecting data sets from motor vehicles. The data collected is then observed based on the vehicle activity that is referred to as an event. There are two types of events, namely; first and second event. The first event correlates the movement and journey of the vehicle while the second event obtains data of the car while the driver is driving. The main idea of this model is to analyse the vehicle movement, speed, location, and mileage and engine events of the vehicle during the trip. The functions of the model are processing CAN bus data through Hadoop, where the data is split into four phases where all the data is sent to the Sqoop where the users can access the data for processing [**?**].

There are four phases included in the processing of the information obtained from the system's remote database. These include the phase of data importation from MySQL to Sqoop. Sqoop is created as an open-source tool and it ensures that data has been extracted from the relational storage bases to the Hadoop [**?**]. The second component is a Hadoop connector known as the Mongo DB and is used as a destination for the inputs and outputs. When loading the collected information from the HDFS to the Hive, it is the Apache Sqoop that performs the function by ensuring that the parallelizing imports are kept across the different mappers. Once data has been loaded into the Hive, the Sqoop's replication table is moved to its own warehouse of data [**?**]. These patterns have been adopted so as to

 doi:10.20944/preprints202309.0148.v1

implement a joint algorithm used in processing the jobs of Hadoop MapReduce after the execution of HiveQL scripts. The reducing side is one which involves a transfer of identical keys to a single reducer.

Xie and Luo [**?**] suggested the use of Trajectory data analysis system design for taxis through big data analysis platform architectural design. The system is split into five layers of the data, namely; source, processing, storage, application and the user access layer. The trajectory data preprocessing model for the design of the taxis, because of the error for positioning the GPS and the exact time of the data collected by the taxis and the mass data involved requires elimination through the processing of the data. The data analysis system for the design for the taxi track, the design method enables the use of SGD algorithm that helps in getting the exact area for the current town [**?**].

Because of the errors which come about based on how the GPS of a system has been positioned, the complexity of road traffic status, and the dynamics of taxi data being real-time, Hadoop has also been used in order to merge the numerous small files hence facilitating the HDFS reprocessing data function. The reason for this is that the large number of small files has a great probability of causing performance degradation. Hence, data is read on a line by line basis within the native folder, then kept once it reaches the 128M mark, thereafter, another later is created to output and the process is repeated up to the point when all files are read [**?**]. The files regarded as errors will thereafter have to be cleaned and processed within the HDFS so as to count the kinds or errors. The unreasonable data is then eliminated when writing graphs through a parallel processing program.

In addition to this, the technological realization, a function included in the system, includes an area of historical filling which involves a Hadoop technology which is supported by the data mart area which includes a massive parallel database known as Greenplumimlement [**?**]. The system uses enamours parallel databases of distributed file storage systems which ensures that real-time computing and a static offline functionality is enabled. One of the most cost-effective models used in predicting road traffic is Apache Spark. The system uses ontology as the main approach because it predicts the status of road traffic congestion [**?**]. In data processing, the unprocessed data is collected using two sensors, these are the passive infrared sensors and ultrasonic sensors where the data analysis processes are conducted.

After the collected sensor data has been processed, it is possible to obtain the count of vehicles, the inbound traffic as well as data on outbound traffic. The only drawback of the system is that the system takes too long to process data [**?**]. In addition to this, it has been noted that the system does not provide an accurate output prediction and the prediction also takes a lot of time.

Prathilothamai, Lakshmi and Viswanthan [**?**] had an input in the car industry by proposing a system that would help motor vehicles in predicting road traffic. The authors advocate for the Apache Spark system to help in dealing with large volumes of data in the car industry to help drivers to navigate easily in lanes. The system focuses on eliminating the problems associated with the current systems such as time-consuming and its inadequacy in giving predictions. The system collects sensor data and it is converted into CSV file and then it is processed in the Apache Spark. The processing is important in providing useful information in making predictions in traffic. The processing of the CSV file id done by the Spark SQ. The Spark SQ detects the number of motor vehicles, human beings and traffic condition. The authors found that this system is ideal n reducing the number of road accidents resulting from the lack of information on the traffic status [**?**].

The system's architecture includes sensor data collected using an Arduino board linked by passive infrared sensors used to collect data. It has to be noted that the PIR sensors are used to detect the interruptions caused by humans at the area moulded by the sensor. Based on this, it has been stated that they achieve sufficiency in predicting real-time condition of traffic (Prathilothamai et al.,2016). The main proof of this sufficiency is that they are also employed in detecting whether or not an individual has entered due to their high sensitivity rates. They also are used as movement detectors by measuring the amounts of infrared light radiating from the objects within the areas of movement.

The system uses the Apache Spark framework whose main component is the S-SQL and is used in query processing of data obtained from the sensors. This is done so that the number of vehicles,

humans and the traffic condition can be determined. Traffic is predicted by classifying it into high, medium and low classes through the employment of an algorithm. In addition to this, a decision tree is used as a predictive model in analysing decisions which can represent decisions explicitly and visually. When processing a decision tree, there is an importation of the MIB library into the Spark in which an ML library is provided. The decision tree will thereafter be used to process the loaded CSV file. When converting sensor data to CSV file, both the PIR and ultrasonic sensors are used. However, to process the data, a connection is created through the use of the Arduino board. In addition to this, the collected sensor data is appropriately loaded due to the employment of the Arduino board.

According to Amini, Gerostathopoulos, and Prehofer [**?**], the invention of big data has resulted in disruptive effects in several sectors with the Intelligent Transportation Systems (ITS) being one of the key sectors affected. They propose that this problem can be sorted by an architecture that is based on a distributed computing platform. The system has a big data analytics engine and control logic. The big data analytics engine alerts the control logic. The system works by sending the average density to the controller from the controller. It works well in a three-lane freeway using Surveillance cameras to sense obstacles on the way.

Real-time traffic control has also been described. In the system, the consumers of the data will make a variety of queries, hence, in the process of platform development, it is important to take into account the variations in the queries. Under the proposed system illustrated in the paper, the queries are divided into three possible groups.

The first is the periodic category which considers the consumers who have to constantly monitor the systems so that they adapt to the ever occurring changes on the information. The second category is that of description. It is one which considers the queries which describe the current state of a system such as the length of signal optimization. The third approach is one which considers time [**?**]. It has been stated that most of the real-time traffic controls will need to be responded to in determinable latencies. However, there are those queries which will not impose any kind of timing demands.

The described platform, in this case, is one which employs Kafka topics to support the system's communication between the incoming information and the traffic systems. The engine of data analytics is one which performs the analysis and control of the logics as defined by the consumers. In addition to this, the system provides an allowance for the users to customize the intervals of time taken to receive the outcomes from the engine of analytics. As data is received, it is processed through a reducer function specified by the user [**?**]. From the system analysis process, it was found that the proposed platform can handle a large bandwidth of receive data due to the efficiency of Kafka. To manage the cases which come about due to the in-memory python computation, there is an option of using the Spark as pre-processor.

Another significant characteristic of the illustrated system is that it provides an allowance to plug-in different sources of data. In adding the sources, the Kafka topics need to be augmented when the data source needs to publish a new type of item [**?**]. A set of python enabled functions are used to ensure that the query specifications in the process of data analytics are well responded to.

Another author Zeng [**?**] additionally had some input on the topic of big data in intelligent Traffic System. The author argues for an architecture of big data technology has several advantages in the System Traffic smart. Big data helps in calculating average speed, determining the lane of a vehicle, monitoring and controlling unroadworthy vehicles. The author points out that the invention of big data has answered data storage, data analysis and data management. Further, the use of big data technology in the traffic system helps the system to deal with a large volume of complex and varied data. Big data achieve these functions through the integration of several systems, models, departments and technologies. Therefore, the author points out that big data technology has transformed intelligent traffic systems by increasing efficiency.

When applying big data on an intelligent traffic system, an intelligent system which combines a variety of systems, models and technology is suffice. The architecture of an intelligent platform of big data used designed for the transport system is a comprehensive scientific, mathematical, economic, as

well as a behaviour intelligent system. It is inclusive of a business layer, that of information publishing, and a layer of data analysis. The main difference between the ITS and the traditionally used system of traffic control is in regards to the intelligence features. It can be noted that the I.T.S can carry out intelligence control based on the condition of the traffic [**?**]. The employed Hadoop ecosystem ensures that a natural advantage has been attained in dealing with big data traffic.

ITS has an ability to report on different traffic conditions. This is done by calculating the flow of traffic of each bayoneting in time intervals such as 5, 10, or 15 minutes. The calculated data is then forwarded to the publishing layer where a report is provided to the most relevant recipient. The most efficient way employed to conduct the statistical analysis is the parallel Hadoop MapReduce model [**?**]. In regards to the average speed indicator, it needs to be noted that its significance is based on the fact that the efficiency of traffic will increase with the traffic speed. This is also calculated through the map reducing process. The third part of the system is that where queries are made in regards to the vehicle's path of travel. This has sufficed in the work of investigating public security as ITS will resolve the problem of a greater demand for manpower. The bayonets are able to both identify and record the plate numbers and save them in the HBase [**?**]. When checking and controlling fake vehicles, the ITS employs big data to identify the fake vehicles because it queries data and calculates the differences of time between the bayonets. A vehicle will be judged as fake in the event the time difference falls below the 5 and 2 minute mark.

Liu [**?**] on his study on the big data analytics architecture for internet –of-vehicles, stated that big data analysis platform can be used to ease congestion on lanes. Liu stresses that the platform needs to be decomposed into three layers namely; infrastructure, data analysis and application layer. The sensors either on the ECUs or roadside detects vehicle information and sends it to the system where it is processing, trajectory forecasting and risk evaluation is done. It has the capacity of solving the challenges of storage, analysis and distribution of information within the transportation sector. Internet – of- vehicle is an important aspect in the industry for the reasons highlighted above. It comprises of perceptive, network and application layers. All the traditional challenges in the transportation industry can be sorted through the wireless sensor network [**?**].

It has been illustrated that the traffic control systems which employ the use of the Spark platform have an infrastructure layer made of two parts. These are the gateway of multi-sensor information system and the roadside sensing nodes. The vehicle gateway is one which includes a GUI with a non–real-time OS on ARM and a real-time computation included on the PAC DSP. On the ECUs, the acquisition and the processing functions are implemented on the non-OS context of microcontroller boards [**?**]. In regards to the wireless vehicle of sensor devices, they are installed inside a car to provide key data on the vehicle as well as the driver's state [**?**]. All of the sensor nodes collectively contribute to the processes of data gathering at various moments to ensure that the most appropriate outputs for decision making are created.

The Lambda architecture provides an integrated solution to the effectiveness of traffic control systems when it is founded in Spark. The mature Hadoop platform of storage and computing is used based on the fact that it stores large sets of data, and the MapReduce is also used due to its flexibility in the function of computing expansion. In addition to the fact that the two tools will ensure that a full view of the data amounts has been generated, the MLib platform provided by Spark can be used to ensure that batch data has been directly analysed, hence facilitating the application development and maintenance. The stream layer has been depicted as one which uses platform of distributed stream processing in which the spark streaming acts as a supplement to the highly latent responses of the batch processing [**?**]. The reason for this is that the Spark Streaming is one which deals with data which increases on a real-time basis so as to provide a low latency view. The view of the storage outputs from both the batch and speed layers have been stated to offer a query function which is of a low latency rate and will also match the real-time processing and batch view.

Lastly, Guo et al., [**?**] mechanism argued that there is a need for a secure mechanism for collecting big data due to the rise of large scale internet of vehicles. IoV has resulted in the interconnection of

numerous vehicles over the internet. The generation of big data in the transportation industry requires other models that would enhance data security during its collection and storage. This model requires all vehicles to be registered in the big data centre to allow the vehicle to connect in the network. This hinders the possibility of unregistered vehicles to access the information. After the first phase single –the sign-on algorithm is used to enhance efficiency in data collection and storage. Once the data is collected, the processing is done using Hadoop architecture to enhance unified control. Generally, the model ensures security on the Internet of vehicles.

6. Conclusion

In the contest of state-of-the-art technologies, the future vision is how to consolidate IoT and big data to help industries to improve operations and to realize a high level of efficiency and cost reduction. This paper introduces basic concepts and characteristics of big data, and Hadoop-based ecosystems. A generic IoT architecture is described and detailed. The paper identifies cloud-based IoT enabling technologies such as software as a service (SaaS) and platform as a service (Paas). A survey of selected integrated solutions for IoT with big data are examined across two applications: Healthcare and Transportation. The solutions show a high utility of Hadoop-based ecosystem and Apache Spark. Various other softwares were also used depending on the use case. Nevertheless, it could be observed that there is no single comprehensive IoT and big data platform and conclusively it depends on requirements and use cases.
Future research could strive to propose a comprehensive architecture for a specific sector. It could also include experimentally comparing the different systems reviewed in this paper.

References

. Saha, A. K., K.A.T.V.; Das, S. Big Data and Internet of Things: A Survey. 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). Greater Noida (UP), India, 2018, pp. 150–156.
. Lu, S.Q.; Xie, G.; Chen, Z.; Han, X. The management of application of big data in internet of thing in environmental protection in China. 2015 IEEE First International Conference on Big Data Computing Service and Applications. IEEE, 2015, pp. 218–222.
. Landset, S.; Khoshgoftaar, T.M.; Richter, A.N.; Hasanin, T. A survey of open source tools for machine learning with big data in the Hadoop ecosystem. *Journal of Big Data* **2015**, *2*, 24.
. Apache Hadoop. https://hadoop.apache.org/, 2019.
. Ounacer, S., T.M.A.A.S.D.A.; Azouazi, M. A New Architecture for Real Time Data Stream Processing. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS* **2017**, *8*, 44–51.
. Machines, I.B. Internet of Things, 2019.
. Karve, R.; Dahiphale, D.; Chhajer, A. Optimizing cloud MapReduce for processing stream data using pipelining. 2011 UKSim 5th European Symposium on Computer Modeling and Simulation. IEEE, 2011, pp. 344–349.
. Satoh, I. MapReduce-based data processing on IoT. 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom). IEEE, 2014, pp. 161–168.
. Jafar, M.J.; Babb, J.S.; Abdullat, A. Emergence of data analytics in the information systems curriculum. *Information Systems Education Journal* **2017**, *15*, 22.
. Jesse, N. Internet of things and big data–the disruption of the value chain and the rise of new software ecosystems. *IFAC-PapersOnLine* **2016**, *49*, 275–282.
. Ahmed, E., Y.I.H.I.A.T.K.I.A.A.I.A.I.M.; Vasilakos, A.V. The role of big data analytics in Internet of Things. *Computer Networks* **2017**, *129*, 459–471.
. Chou, S.C.; Yang, C.T.; Jiang, F.C.; Chang, C.H. The Implementation of a Data-Accessing Platform Built from Big Data Warehouse of Electric Loads. *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* **2018**, *2*, 87–92. doi:10.1109/COMPSAC.2018.11821062.

. Maarala, A.I.; Rautiainen, M.; Salmi, M.; Pirttikangas, S.; Riekki, J. Low latency analytics for streaming traffic data with Apache Spark. 2015 IEEE International Conference on Big Data (Big Data). IEEE, 2015, pp. 2855–2858.
. Apache Hive TM. http://hive.apache.org/, 2019.
. Ta-Shma, P., A.A.G.G.G.H.G.C.F.; Moessner, K. An ingestion and analytics architecture for iot applied to smart city use cases. *IEEE Internet of Things Journal* **2017**, *5*, 765–774.
. Miorandi, D., S.S.D.P.F.; Chlamtac, I. Internet of things: Vision, applications and research challenges. *Ad hoc networks* **2012**, *10*, 1497–1516.
. Malik, V.; Singh, S. Cloud, Big Data & IoT: Risk Management. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). Faridabad, India, 2019, pp. 258–262.
. Sassi, M. S. H., J.F.G.; Fourati, L.C. A New Architecture for Cognitive Internet of Things and Big Data. *Procedia Computer Science* **2019**, *159*, 534–543.
. L. Zhu, F. R. Yu, Y.W.B.N.; Tang, T. Big Data Analytics in Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems* **2019**, *20*, 383–398.
. Yadav, P.; Vishwakarma, S. Application of Internet of Things and Big Data towards a Smart City. 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU). Bhimtal, 2018, pp. 1–5.
. Ma, H.D. Internet of things: Objectives and scientific challenges. *Journal of Computer Science and Technology* **2011**, *26*, 919–924.
. Collins, E. Intersection of the cloud and big data. *IEEE Cloud Computing* **2014**, *1*, 84–85. doi:10.1109/MCC.2014.14.
. Elshawi, R.; Sakr, S.; Talia, D.; Trunfio, P. Big data systems meet machine learning challenges: Towards big data science as a service. *Big data research* **2018**, *14*, 1–11. doi:10.1007/s41065-017-0092-2.
. Rathore, M. M., A.A.; Paul, A. The Internet of Things based medical emergency management using Hadoop ecosystem. 2015 IEEE SENSORS. IEEE, 2015, pp. 1–4.
. Taher, N. C., M.I.A.N.; El-Mawass, N. An IoT-Cloud Based Solution for Real-Time and Batch Processing of Big Data: Application in Healthcare. 2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART). IEEE, 2019, pp. 1–8.
. Chhowa, T.T.; Rahman, M.A.; Paul, A.K.; Ahmmed, R. A Narrative Analysis on Deep Learning in IoT based Medical Big Data Analysis with Future Perspectives. *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)* **2019**, pp. 1–6. doi:10.1109/ECCE.2019.8817489.
. A. Zameer, M. Saqib, V.R.N.; Ahmed, I. IoT and Big Data for Decreasing Mortality rate in Accidents and Critical illnesses. 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC). Muscat, Oman, 2019, pp. 1–5.
. Rei, J., B.C.; Sousa, A. Assessment of an IoT Platform for Data Collection and Analysis for Medical Sensors. 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC). Philadelphia, PA, 2018, pp. 405–411.
. Mezghani, E., E.E.; Drira, K. A model-driven methodology for the design of autonomic and cognitive IoT-based systems: Application to healthcare. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2017**, *1*, 224–234.
. Ma'arif, M. R., P.A.S.C.B.; Cahyo, P.W. The Design of Cost Efficient Health Monitoring System based on Internet of Things and Big Data. *2018 International Conference on Information and Communication Technology Convergence (ICTC)* **2018**, pp. 52–57.
. Nkenyereye, L.; Jang, J.W. Integration of big data for querying CAN bus data from connected car. 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, 2017, pp. 946–950.
. Xie, J.; Luo, J. Construction for the city taxi trajectory data analysis system by Hadoop platform. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)( ). IEEE, 2017, pp. 527–531.
. Prathilothamai, M., L.A.M.S.; Viswanthan, D. Cost Effective Road Traffic Prediction Model using Apache Spark. *9 (May)* **2016**.
. Amini, S.; Gerostathopoulos, I.; Prehofer, C. Big data analytics architecture for real-time traffic control. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)* **2017**, pp. 710–715. doi:10.1109/MTITS.2017.8016503.

. Zeng, G. Application of big data in intelligent traffic system. *IOSR Journal of Computer Engineering* **2015**, *17*, 01–04.
. Liu, D. Big Data Analytics Architecture for Internet-of-Vehicles Based on the Spark. 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS). IEEE, 2018, pp. 13–16.
. Guo, L.; Dong, M.; Ota, K.; Li, Q.; Ye, T.; Wu, J.; Li, J. A secure mechanism for big data collection in large scale Internet of vehicle. *IEEE Internet of Things Journal* **2017**, *4*, 601–610. doi:10.1109/JIOT.2016.2637144.