ISSN 2278-3091
Volume 9, No.1, January – February 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse118912020.pdf
https://doi.org/10.30534/ijatcse/2020/118912020

Towards a Reference Big Data architecture for sustainable smart cities

Saida EL MENDILI [1], Younès EL BOUZEKRI EL IDRISSI[1], Nabil HMINA[1]
[1]Systems Engineering Laboratory, National School of Applied Sciences, Ibn tofail university, Kenitra, Morocco
elmendili.saida@uit.ac.ma
y.elbouzkeri@gmail.com
hmina@univ-ibntofail.ac.ma

ABSTRACT
The creation of smart cities aims to reduce the problems posed by the continuous evolution of population density and urbanization. Smart City applications produce a huge amount of data every day. Thus, the knowledge of these large data in a context of urban and intelligent decision-making has become an issue for current systems. Large data analysis frameworks offer significant inventive potential in the new area of the smart community. This paper propose a new architecture for large data analysis for smart cities called "BIG DATA ANALYTICS FRAMEWORK FOR SMART CITY (**BDAFSC**)", The proposed framework specifically addresses a conceptual and technological model by creating several layers of abstraction. The proposed architecture is generic and can be applied to a wide range of smart city use cases.

Key words: Smart city, data analytics, spark, Hadoop, Big data, Internet of things, Urban data, structured data, semi-structured data, unstructured data, Spark streaming.

1. INTRODUCTION

Today, the Internet of Things is one of the emerging technologies, it is clear that in the coming years, all the things on this earth will be connected to each other and we will live in a connected world. Every individual must be informed of this technology that is invading the world where every object, device or person will be connected to each other and will live a better organized life than today's, targeting sustainable economic growth and a better quality of life for its inhabitants and visitors. The Internet of Things is a new technology for the smart city that interconnects different digital devices through the Internet, offering many innovative tools from academia to industry. The smart city is an ubiquitous conceptual change that has transformed the entire landscape through the use of information and communication technologies (ICT).
The rapid growth of information and communication technologies offers opportunities to many societal services, providing users with techniques to access different types of systems [33].
Smart City applications create a large amount of data that makes up large volumes of data. The extraction of hidden information and relationships from these large data is a new trend in information systems, to provide better services to citizens and support the decision-making process. The proposed framework deals specifically with the conceptual and the technological model by creating multiple layers of abstraction, more, the machine learning part is implicitly included, in the form of a new optimized algorithm that aims to generalize a set of rules from historical data in order to describe the reel data generation process, predict future events and allow decision-making process, therefore extracting valuable insights to plan for any expansion in smart city services, resources, or areas.

We propose a new architecture for ingestion and historical data analysis to provide meaningful and useful information for real-time analysis. in our proposal, historical (batch) analytics will improve the quality of real-time analytics on IoT data. We deploy our architecture using open source elements optimized for large data applications such as spark and Hadoop framework.

This section is an introductory section about the subjects and motive for this paper, the rest is organized as follows: the background literature and Related works in this field are reviewed and reported in Section 2, architectural specification, the Data Flow Diagram, the conceptual and the technological view are reviewed, explained, briefed and illustrated in Section 3. and the conclusion together with future research trends are presented in Section 4.

2. BACKGROUND LITERATURE AND RELATED WORKS

In this context, multiple architectures are proposed, focusing only on sensor data, social networks or intelligent applications. The study of these works shows significant results, but they are incomplete because the authors propose an architecture adapted only to a subject related to a data source, while the data sources in an intelligent city are diverse. In addition, the authors are not always sufficiently detailed on how the technologies used will work and communicate with each other in a large data environment, as well as on the availability of data, despite all the constraints mentioned above, it was possible to find some work addressing the same research question as us.
The Lambda architecture was proposed by [1] to address this, and provides a scalable and fault tolerant architecture for

processing both real-time and historical data in an integrated fashion. The purpose of this architecture was to present a software design pattern, the lambda architecture unifies online and batch processing within a single framework. The pattern is suited to applications where there are time delays in data collection and availability through dashboards, requiring data validity for online processing as it arrives. The pattern also allows for batch processing for older data sets to find behavioral patterns as per user needs.

The purpose of this architecture is to analyze very large amounts of data upon receipt in an efficient, timely and fail-safe manner. Emphasis was placed on the speed of OLAP (Online Analytical Processing) style calculations, such as viewing web pages and analyzingclickpaths. It was not designed to make event-based decisions or react to events as they occur [2]. It includes batch, rate and service processing layers, which must be synchronized to work closely together, and it is complex and difficult to deploy and maintain [3].

The work of [4] presents a model: the BASIS architecture, facilitates the undervaluation of each of the following elements and its contribution to the study of large data for smart cities, and aims to reduce the uncertainty factor associated with the role of several large data technologies and their interdependence in smart cities, in order to address the challenges of extracting, storing, processing and analyzing the huge volume and variety of data found there at high speed.

BASIS contributes to the scientific development of large data architectures for smart cities with unprecedented multiple levels of detail, including, for example, concerns about open data, data mining on distributed environments and their administration, monitoring and security. It should be noted that most parts of the architecture have already been validated with demonstration cases, but that the layers are not clearly demonstrated due to the distinction between technology and infrastructure. As the infrastructure layer concerns aspects of physical hardware design for interacting with the outside world that could be considered as a functionality of the technology layer, although BASIS is a complete architecture for large data analysis, it does not show the specific aspects of design concerning smart cities.

The authors in[5], propose the RADICAL platform which is a service-oriented architecture (SOA) based platform that allows the collection and analysis of sensitive and social IoT (Internet of Things) data to offer smart cities a variant of value-added services.

IoT data is stored in the RADICAL repository (MySQL database) via the corresponding application programming interface (API). The data relating to the device are recorded in the form of observation and measurement values. Observations correspond to reported general IDE events, while measurements correspond to more specific metrics included in an observation (e.g. CO2 measurements). On the other hand, SN data is accessible in real time from the underlying SN adapters using communication with the APIs of the respective networks.

In addition to the main platform, RADICAL offers a set of application management modules that allow end users to make better use of the platform, for example by installing IoT devices and using the MySQL database as the main storage repository. Although RADICAL is complete, it has some disadvantages: the use of the MySQL database as the main repository is a restrictive factor when it comes to large data and the inability to store the results of analyses for later analysis or reference.

Recently, the authors in[9], propose the "Smart City Data Analytics Panel - SCDAP". Before proposing his own architecture, the author studied the different existing solutions, namely: BASIS, SWIFT and Radical, in studying these platforms, several deficiencies were identified, as mentioned above, in order to better address these deficiencies, the authors propose a new framework that introduces new functionalities to the main data analysis frameworks represented in the management and aggregation of data models. The value of the proposed framework is discussed in comparison to traditional knowledge discovery approaches, which introduces new features to large data analysis frameworks for smart city applications, the main feature of this architecture is limited to the Apache Hadoop suite as an underlying layer for data storage and management. The separation between the functionality of the SCDAP and the underlying data storage and management layer will add to the generality of the SCDAP and its ability to deal with many other platforms. However, the framework has several failures:

- The author did not take into account the functionalities of the data quality layer that redefine the way data supervision is performed. Data quality management involves adding additional functionality at each step with continuous quality control and monitoring to avoid quality failures during all phases of the proposed architecture.
- The performance of the SCDAP framework is not tested.
- The development of efficient algorithms for processing and extraction is not explained.

Unlike existing solutions, our architecture presents an hybrid approach to solve the challenge of processing massive amounts of historical data, while at the same time ingesting and analyzing real-time data at a high rate. The duplicity of event processing frameworks for real-time data and batch processing frameworks for archived data has resulted in the presence of multiple independent systems that analyze the same data.

The architecture encompasses several categories of learning algorithms, which are designed with the goal to achieve robustness, adaptiveness, and computational efficiency.

Learning algorithms depends on the decision-making tasks: classification, association rules, anomaly detection, prediction. Furthermore, our proposal encompasses data quality requirements by adding features at each stage with continuous quality control to avoid quality issues during all phases of the data cycle. Thus, this approach ensures the security and confidentiality by the implementation of a security layer that provides these features in each stage of the data cycle according to the components used (eg: Hadoop uses secure). Our approach is practical, scalable and its development, deployment and maintenance are inexpensive. it will be practically implemented and tested from the real world in the transport domains, to provide experimental highlights.

3. BDAFSC : ARCHITECTURAL SPECIFICATION

3.1. The Analytical Architecture and Implementation Model

In order to provide an intelligent and sustainable space, smart cities need to process very large amounts of data, taking advantage of the opportunities associated with the vast network of connected devices currently called IoT [3]. In this context, data are drawn from multiple sources and the ability to integrate, process and analyze this data has a critical influence on the availability of new information services. With these necessities come significant challenges that can be exceeded with the adoption of an adequate Big Data architecture, leading to the proposal of BDAFSC.

The proposed architecture meets the requirements of effective real-time processing as well as the challenges imposed by the process of historical data analysis from intelligent services. For the smart city concept to be effective, the data generated must be analyzed in advance to allow real-time responses to new situations that typically require real-time response to events based on knowledge of past events.

Historical knowledge is essential to understand what behavior is expected and what is an anomaly. Despite its ease of use, our architecture is developed to process very large amounts of historical data and can detect complex events in near-real time using large data analysis systems and automatic training methods.

The proposed framework is dealing particularly with a data flow architecture from the source to the decision-makers end user.

The BDAFSC is a novel FRAMEWORK Big Data for Smart Cities built on a fundamental design principle, it gives a high-level view of the various components in a solution and identify how they fit together to solve our problem.

To better understand the proposal architecture, we will distinguish between the Data Flow Diagram, the conceptual view, the flowchart diagram and the technological view. The strict separation between these presentations will allow us to better understand the proposed relationship model between different entities in a smart city context.

3.2. The Data Flow Diagram

Figure 1 shows the BDAFSC data flow diagram, which shows us, how data moves through an information system, which external processes or entities create or consume the data, and where it is stored, in our case, the data flow diagram (DFD) presents the method of analysis and design of structured systems to show how dataflows through our architectural system including processes.

Data acquisition defines the process of collecting data from an intelligent city environment via a data transmission mechanism. In cases where real-time data is exploited without taking advantage of historical data, it can be said that real-time flows are normal, although historical data can generally provide information that is important for making a real-time break. To do this, using automatic learning algorithms, a repository model was used to manage the data that is extracted, in which the resulting data analysis models can be retained, extracted with the metadata that concerns them for future surveys or reused. the model is stored in this repository.

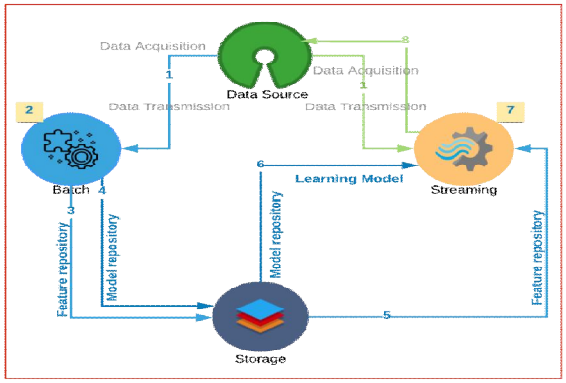

Figure 1: Data Flow Diagram of BDAFSC

3.3. The conceptual view

Figure 2 presents the conceptual architecture of the proposal architecture, is a 4 layer architecture including:A) Data source Layer, B) Data collection Layer, C) Data Management Layer and D)Data Application Layer :

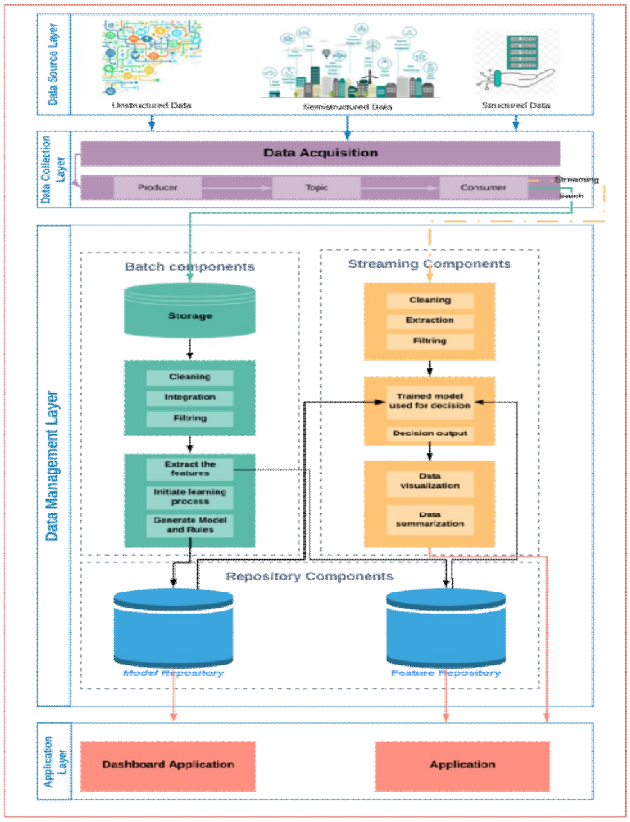

Figure 2: Conceptual view of BDAFSC

A. Data source Layer:

The diversity of data sources and the variety of formats of the data generated is a big challenge of data collection, according to [3], A smart city, represents a rich environment with multiple potential data sources that can generate two types of data:

- Data with low speed data and concurrency, integrating (CSV, TXT, JSON, XML) and programmed readings

(periodic readings from databases or historical data from the Web, such as newsfeeds, for example);
- Data with high speed and concurrency, representing data flows from the web (tweets, blog posts) or electronic devices (smart meters and other sensors), smart phones, weather stations, geo-localizable devices).
As shown in figure 3, data can be:

- **Structured data:**

Structured data is data that has been organized into a formatted repository[6], typically a database, so that its elements can be made addressable for more efficient processing and analysis.

- **Unstructured data:**

Unstructured data [7] is data that is not saved in a fixed recording format. Such as documents, social media streams and digital images and videos. Unstructured data is data that is not found in the rows and columns of a traditional database, it is the opposite of structured data.

- **Semi-structured data:**

Semi-structured data [8] is data that has not been organized into a specialized repository, as is the case in a database, but which nevertheless includes related information, such as metadata, which makes them easier to process than raw data.
Semi-structured data is data that is of the intermediate form, it is not organized in a way that will facilitate access and analysis. However, some information may be associated with them, such as metadata tags, which allow the addressing of the elements they contain.

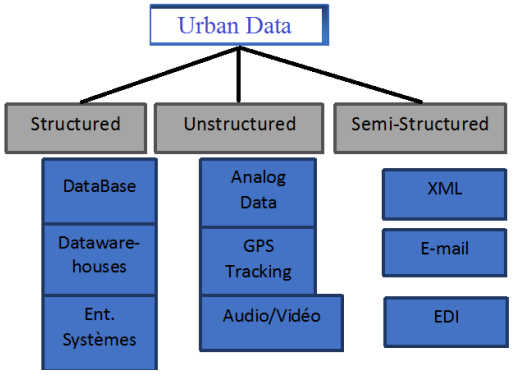

Figure 3: Urban Data**.**

Several data loading and retrieval devices are included in the technical architecture to address the main challenge of data collection, namely managing the diversity of data formats.

B. Data Collection layer:

In a smart city, data can be collected from various sources, for example, IoT testbeds, social networks, sensors, video surveillance systems, or other shared open data like city maps, bus time schedule information, location of restaurants etc. The biggest challenge for data collection is to deal with the format diversity of different data sources. In order to provide a unified interface for collecting data, we design many components which has the wire or wireless connection with anumber of sensors and responds to data management layer requests. In this layer, data is collected for the external world to the proposed framework. The proposed design of this layer enables the ability to collect stream data and batch data from the external world by the Data acquisition and data transmission mechanisms.

C. Data Management Layer:

This is the core data layer mechanism that provides all the data management layer functionalities from data gaining to features and extraction model.
Once the data is transmitted from the upper layer, at this stage, it must be pre-processed. The data pre-processing is an import step in the BIG DATA process. Raw data is always at risk to noise, missing values, and inconsistency. The quality of data affects the mining results. Data pre-processing deals with the preparation and transformation of the initial dataset. Data pre-processing methods are divided into 3 categories:

- **Data Cleaning**
- **Data Integration**
- **Data filtering**

To do this, several types of algorithms incorporate our architecture repository to counter the problems associated with the analysis of the smart data. these algorithms perform a series of operations to overcome the problems associated with data quality, especially:

- Algorithm for Addressing Data Cleaning Problem
- Algorithm for Addressing Data Integration Problem
- Algorithm for Addressing the Data filtering Problem

Furthermore, this layer support: streaming and batch components. It provides two important functionalities that differentiate the proposal framework, namely: model and feature repository where knowledge and insights are extracted.
This repository will learn from the latest data flow, in order to meet real-time computing requirements during online deployment. So, the learning algorithms are designed to achieve the robustness and efficiency of calculations.
Our architecture faces the challenge of handling massive amounts of historical data while ingesting and analyzing real-time data at high throughput. The dichotomy between event processing frameworks for real-time data and batch processing frameworks for archived data has led to the proposal of this framework based on the integration of several independent systems by analyzing the same data, for example to identify failures, our system must first learn a normal behavior of historical data.
The data is ingested by the message broker in a data storage structure for permanent storage. The data can then be retrieved and analyzed using long-term batch calculations, for example by applying our machine learning algorithms. The result of this analysis may affect the behavior of the real-time event processing framework. The batch flows will operate independently of the real-time flows to form the proposal's predictive model.

D. Data Application Layer

After data collection and management, the data analysis models generated by the proposed framework will be used by several Smart City applications to make decisions and improve citizens' quality of life.

It is essential to visualize the information obtained in such a way that the user can easily assimilate it, this objective is achieved by using data post-processing techniques, which are categorized into two categories: Visualization and summary of data.

E. Data Quality Layer

The importance of data quality in the life cycle of large data redefines the way data supervision is approached [10]. Data quality management is about adding more functionality at each step with continuous quality control and monitoring to ensure that there are no quality issues during all phases of the life cycle. When assessing the quality of large data, attention is paid to properties such as performance, value and cost.
Data quality problems occur when quality criteria are not met for data values [11]. These problems are due to several other processes or factors that have occurred at different levels:
- **Data sources**: lack of security, trust, data copying, inconsistency, multiple sources and data domain,
- **The generation level:** human data entry, sensor reading, social media, unstructured data, and missing values,
- **The level of the process** and/or application (acquisition: collection, transmission).
Data pre-processing improves data quality by performing many tasks and activities such as data transformation, integration, merging and standardization.
The authors in [12],[13] listed several causes of poor data that affect data quality and listed elements that have an impact on data quality and its associated dimensions. In [14],[13], the authors focused on a compilation of causes of poor data classified according to data quality dimensions, level of granularity and type of data source, while emphasizing causality mapping.

F. Data security Layer:

The diversity of data sources and formats, continuous data distribution and infrastructure can lead to unique security vulnerabilities [31]. The Cloud Security Alliance has divided data security and privacy challenges into four broad categories: security of all infrastructures, data protection, data management, data integrity and reactive security [32]. Infrastructure security consists of securing distributed programming and security practices in non-relational data warehouses. Data confidentiality refers to the protection of privacy by preserving analyses, the encrypted data center and granular access control. Data management involves the secure storage of data and transaction logs, audit and data source. In addition, integrity and reactive security include real-time validation, filtering and monitoring. Based on the proposed questions, authorization and authentication mechanisms must be put in place for users and applications, and data encryption and masking must be implemented for rest and data flow.

3.4. The flowchart Diagram

The explicit flowchart describes in detail the visual representation of the sequence of steps and decision-making processes required to execute the processes of our architecture. This allows anyone to view the diagram and logically follow the process from beginning to end. The diagram of the diagram is described in Figure 4.

Figure 4: flowchart Diagram of BDAFSC

3.5. The technological view

Although the theoretical view provides a complete and consistent picture of how intelligent data can be extracted, stored, processed and made available in smart cities, it is therefore important to match the technologies needed to instantiate the BDAFSC. The technological detail that will be presented in this section represents the contribution of the proposed architecture, Figure 5 illustrate the technological layer by highlighting the different technologies of acquisition, transmission, pre-processing, storage, processing and analysis. The choice of technologies used is most often generic and can be applied to various types of use case.

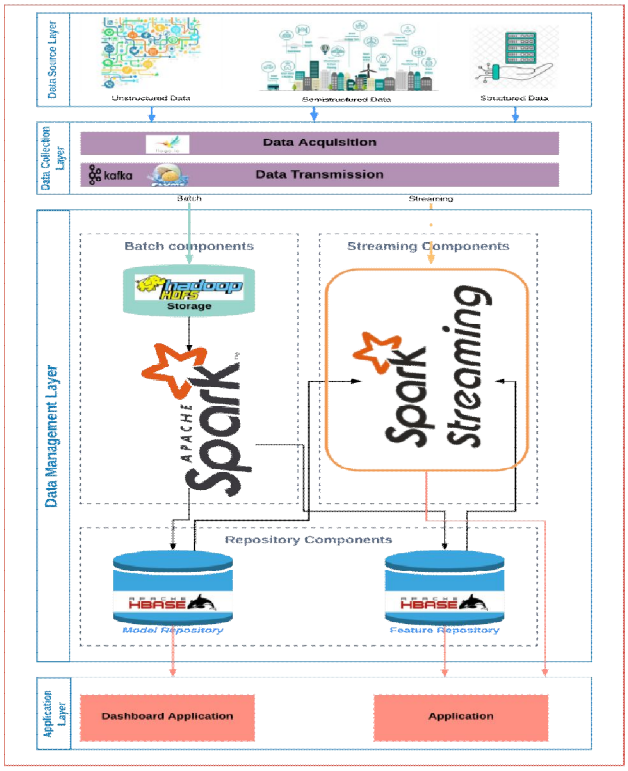

Figure 5: The technological view of BDAFSC

A. Data Acquisition - KURA:

Across the various architectures to large data processing systems, the core of data collection is the collection of data from distributed information sources in order to store them in a large storage that can be extended and adapted to data. To achieve this objective, three main elements are essential [15]:
-Protocols that allow the collection of information for Apa distributed data sources of all types (unstructured, semi-structured, structured).
-The frameworks with which data are collected from distributed sources using different protocols
-Technologies that allow the persistent storage of datathat is recovered by frameworks In the platforms, devices are connected to centralized server that orchestrate the communication among devices. However, the platforms cannot make real-time decisions quickly enough when they are required to traverse a series of networks to access computing resources. There are also security concerns around transmitting sensitive data across networks and into the cloud [16].

For these reasons, IoT peripheral platforms such as Kura [17], OpenHAB[18], Red-Node[19] and Flogo[20] appear. Kura is one of the popular IoT Edge Platform which is composed of Apache Cameland the Eclipse.
Kura will focus on IoT Gateway and offer a set of java and OSGi services. On the platform, developers must execute their own source code without the help of a visual designer or scripting tool. Thus, the developer gets more flexibility for customization, but has a much higher learning curve.

B. Data transmission-Kafka, Flume, Zookeeper [22]:

The data collection unit needs a distributed, reliable and available framework that can handle large amounts of data. This model is mainly implemented by a Flume distributed cluster, which allows multi-source acquisition and localized storage. Currently, Flume has evolved from the OG version to the NG version. Flume uses an exchange-based data transfer method to ensure the reliability of transaction transfer. In addition, Flume has a very high extension that supports multi-level extensions. The topology of the data collection module is composed of one or more agents according to specific logical needs. This is why Flume is much better than other log collection frames.
The data transmitted from the data collection part (Flume cluster) are linked to the producer side of Kafka through Kafka Sink. Kafka will adjust it through its own configuration file. Number of backups, number of service nodes, number of partitions and load balancing to ensure a stable and efficient transmission process. The mainstream part of Kafka connects to Spark Streaming, the central part of the data processing part.Kafka is a messaging system that adopts thepublication/subscription mode, developed by Java, supports parallel loading of Hadoop and Spark data, and provides a reliable solution for real-time processing functions of offline frameworks such as Hadoop and Spark. One of the advanced aspects of Kafka is its excellent flow capacity. Kafka uses Zookeeper to complete the data management, which compensates for the pointer problem.

C. Preprocessing - Spark [21]:

Apache Spark can alleviate key challenges of data pre-processing, iterative algorithms, interactive analytics and operational analytics among others. With Apache Spark, data can be processed through a more general directed acyclic graph (DAG) of operators using rich sets of transformations and actions. It automatically distributes the data across the cluster and parallelizes the required operations. It supports a variety of transformations which make data pre-processing easier especially when it is becoming more difficult to examine big datasets. On the other hand, getting valuable insights from big data requires experimentation on different phases to select the right features, methods, parameters and evaluation metrics.

D. Storage HDFS-HBase [30]:

By design, HDFS is perfectly tolerant to disturbances and allows fast data transfer between nodes, even in the event of a system failure. HBase is a non-relational and open source Not-Only-SQL database that runs on Hadoop. HBase is part of the CP CAP (Consistency, Availability, and Partition Tolerance) theorem.

HDFS is best suited for batch analysis. However, one of its biggest shortcomings is its inability to perform real-time

analyses, which is in line with IT industry trends. HBase, on the other hand, can handle large datasets and is not suitable for batch analysis. Instead, it is used to write/read Hadoop data in real time.

E. Batch components- Spark:

Apache Spark is a general analysis engine that can processlarge amounts of data from various data sources and has gained popularity on significant It works particularly well for multi-pass applications that include many machine learning algorithms [23]. Spark maintains an abstraction called Resilient Distributed Datasets (RDDs) that can be stored in memory without the need for replication and is always fault-tolerant.

For model learning, we use Spark MLlib[24] which is the Sparks library for machine learning. Its goal is to make automatic learning practical, scalable and easy to use. Spark MLlib consists of common machine learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as lower level optimization primitives and higher-level pipeline APIs.

F. Streaming Components – Spark streaming:

The data can be streamed in real time as part of Apache Spark streaming from various sources such as Kafka, flume, HDFS, etc. A receiver must be instantiated and connected to the streaming source to initiate the data stream. A receiver can only broadcast data from a single input source, and if we have several flow sources, we can group them together so that they can be treated as a single flow [29]. Once the receiver starts receiving data from the streaming source, the spark stores the data in a series of RDDs delimited by a specified time window. After this time, the data is sent to the spark core for processing. To start a Spark Streaming job, it needs at least two kernels, one that receives the data as a flow and one that processes it.

There are two main cases where Spark Streaming and MLlib can be used together. Machine learning models generated offline with MLlib can be applied to streaming data (offline training, online prediction). On the other hand, automatic learning models can be driven from labelled data flows (i.e. online training and prediction). A reference application that uses Spark Streaming with Spark MLlibis Twitter Streaming Language Classifier [25]. Another is a platform for large-scale neuroscience [26]. Spark Streaming is integrated with MLlib to develop streaming machine learning algorithms and perform online analysis during experiments. In addition, Spark MLlib supports some streaming machine learning algorithms such as Streaming Linear Regression and K-means Streaming [27].

As part of this architecture, we use machine learning techniques in particular, trying to collect accurate data and provide accurate information to end users. The simulation results show that our proposals could achieve a high level of QoE for the smart city.

4. CONCLUSION

Although the proposed BDAFSC introduces new functionalities to large data analysis frameworks for smart city applications, it is still relevant to demonstrate its application to real problems to explain its contribution to optimization and decision-making for large smart city applications. This approach will therefore be implemented and tested in practice on a case study of the use of real-world smart cities and we will present the experimental strengths in our future paper.

REFERENCES

1. **Simplifying the (complex) Lambda architecture**. [Online]. Available: https://voltdb.com/blog/simplifying-complex-lambda-architecture
2. **Questioning the Lambda Architecture**. [Online].Available: https://www.oreilly.com/ideas/questioning-the-lambda-architecture http://voltdb.github.io/app-fastdata/
3. Costa C and Santos **MY., "BASIS: A big data architecture for smart cities,"** in In SAI Computing Conference (SAI), London, United Kingdom, 2016. https://doi.org/10.1109/SAI.2016.7556139
4. Psomakelis E, Aisopos F, Litke A and Tserpes K, Karda, "**Big IoT and Social Networking Data for Smart Cities Algorithmic Improvements on Big Data Analysis in the Context of RADICAL City Applications**," in arXiv preprint arXiv:1607.00509 , 2016.
5. Jara, A. J., Bocchi, Y., &Genoud, D. (2013). **Determining Human Dynamics through the Internet of Things**. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). doi:10.1109/wi-iat.2013.161
6. S, EL MENDILI.,Y,EL BOUZEKRI., and N,HMINA., **Big Data Processing Platform on Intelligent Transportation Systems, 2019,** International Journal of Advanced Trends in Computer Science and Engineering, 8(4), July- August 2019, 1099 - 1109. https://doi.org/10.30534/ijatcse/2019/16842019
7. https://www.webopedia.com/TERM/U/unstructured_data.html
8. https://www.lemagit.fr/definition/Donnees-semi-structurees
9. Osman, A. M. S. (2019). **A novel big data analytics framework for smart cities.** Future Generation Computer Systems.https://doi.org/10.1016/j.future.2018.06.046
10. Taleb, Ikbal&Serhani, Mohamed &Dssouli, Rachida. (2018). **Big Data Quality: A Survey**. 10.1109/BigDataCongress.2018.00029.
11. C. Fürber and M. Hepp, "**Towards a Vocabulary for Data QualityManagement in Semantic Web Architectures**," in Proceedings of the1st International Workshop on Linked Web Data Management, NewYork, NY, USA, 2011, pp. 1–8.

12. M. Chen, M. Song, J. Han, and E. Haihong, "**Survey on data quality**,"in 2012 World Congress on Information and CommunicationTechnologies (WICT), 2012, pp. 1009–1013.
13. N. Laranjeiro, S. N. Soydemir, and J. Bernardino, "**A Survey on DataQuality: Classifying Poor Data,"** in 2015 IEEE 21st Pacific RimInternational Symposium on Dependable Computing (PRDC), 2015,pp. 179–188.
14. F. Sidi, P. H. ShariatPanahy, L. S. Affendey, M. A. Jabar, H. Ibrahim,and A. Mustapha, "**Data quality: A survey of data quality dimensions,**"in 2012 International Conference on Information Retrieval KnowledgeManagement (CAMP), 2012, pp. 300–304.
15. @Inbook{Lyko2016,author="Lyko, KlausandNitzschke, MarcusandNgongaNgomo,Axel-Cyrille",editor="Cavanillas, Jos{\'e} Mar{\'i}aand Curry, EdwardandWahlster, Wolfgang",title**="Big dataAcquisition",**bookTitle="New Horizons for aData-Driven Economy: A Roadmap for Usage and Exploitation of Big Data inEurope",year="2016",publisher="Springer InternationalPublishing",address="Cham",pages="39--61",
16. Sameer Ahmed. 2018. Edge Computing: **Unlocking theBusiness Value of the IoT.** (2018).https://dzone.com/articles/edge-computing-unlocking-the-business-value-of-the
17. Eclipse. 2018. Kura. (2018).https://www.eclipse.org/kura/
18. Manuel Raffel. 2014. openHAB - **Empowering the SmartHomeHistory , Concepts ,** Examples. (2014).
19. Node-RED, "Node-RED: A visual tool for wiring the Internet of Things," http://nodered.org//, 2016, [Online; accessed 6-May-2016].
20. TIBCO. 2018. Flogo. (2018). http://www.flogo.io/
21. **Big data analytics on Apache SparkSalman Salloum1** Ruslan Dautov1 · Xiaojun Chen1 · Patrick Xiaogang Peng1 Joshua Zhexue Huang1Int J Data Sci Anal (2016) 1:145–164DOI 10.1007/s41060-016-0027-9.
22. Xie, W., Li, P., & Xu, H. (2018). **Architecture and Implementation of Real-Time Analysis System Based on Cold Chain Data.** Advances in Intelligent Systems and Computing Complex, Intelligent, and Software Intensive Systems, 497–505. doi: 10.1007/978-3-319-93659-8_44.
23. M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley,M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributeddatasets: **A fault-tolerant abstraction for in-memory cluster computing**,"inProceedings of the 9th USENIX Conference on NetworkedSystems Design and Implementation, ser. NSDI'12. Berkeley, CA,USA: USENIX Association, 2012, pp. 2–2. [Online].Available:http://dl.acm.org/citation.cfm?id=2228298.228301.
24. X. Meng, J. Bradley, B. Yuvaz, E. Sparks, S. Venkataraman, D. Liu,J. Freeman, D. Tsai, M. Amde, S. Owen et al., "Mllib: **Machine learningin apache spark**," JMLR, vol. 17, no. 34, pp. 1–7, 2016.
25. Databricks: **Databricks spark reference applications**. http://tinyurl.com/gwzkqxr (2015).
26. I18 Freeman, J.: **A platform for large-scale neuroscience**. Presentation.https://spark-summit.org/2014/talk/A-platform-for-largescale-neuroscience (2014).
27. Freeman, J.: **Introducing streaming k-means in spark** 1.2.https://databricks.com/blog/2015/01/28/introducing-streamingk-means-in-spark-1-2.html (2015).
28. Salloum, Salman, et al. **"Big data analytics on Apache Spark."** International Journal of Data Science and Analytics 1.3-4 (2016): 145-164.
29. Grulich, Philipp M., and Olaf Zukunft. "**Bringing Big Data into the Car: Does it Scale?" Big Data Innovations and Applications (Innovate-Data)**, 2017 International Conference on. IEEE, 2017.
30. https://www.kdnuggets.com/2017/05/hdfs-hbase-need-know.html
31. Sinanc, Duygu and Terzi, Ramazan and Sagiroglu, Seref. **"Bringing A survey on security and privacy issues in big data**, 2015, in proceedings of International Conference : ICITST. doi = 10.1109, pp. 202-207.
32. **Cloud Security Alliance Big Data Working Group**, "Expanded Top Ten Big Data Security and Privacy Challenges", April 2013.
33. Anchal , Pooja Mittal, **Data Mining Techniques for IoT enabled Smart Parking Environment : Survey**, et al., International Journal of Advanced Trends in Computer Science and Engineering, 8(4), July- August 2019.