International Journal of Innovation in Computer Science and Information Technology
Vol.3, NO.1 , P: 3 -16
Received : 8 January 2021
Accepted : 19 March 2021

A survey of the use of big data technologies in smart city implementation

Ali NematiNia*
Master of Software, ICT Organization of Karaj Municipality
Ali.NematiNia@gmail.com
Jila Malayeri
Master of IT Management, ICT Organization of Karaj Municipality
J.Malayeri@gmail.com

Abstract

Implementing a smart city has recently become one of the most widely used concepts in scientific circles and an essential issue among governments. In a smart city, the use of digital technologies can be considered as a better public service for the inhabitants of this city and better use of resources. So far, various studies have been conducted on the technologies used in smart cities. In this paper, we will review the infrastructure models and communication protocols of software in cities and the use of big data techniques in the implementation of smart cities and considerations of the use of big data technology. To provide a better insight into the advancement of researchers, managers and urban planning.

Keywords: Smart city, big data, Hadoop ecosystem.

1- Introduction

The smart city uses technology, people and processes to improve every aspect of the city's performance and improve services to its residents. Technology empowers the smart city, but it is the data that leads to new insights and services. The smart city has advantages such as efficient use of resources, increasing the quality of life and achieving a high level of transparency for citizens, and uses new processes for the development and management of the city. These processes use a set of new technologies that allow citizens to freely access information and use urban information and applications. As cities evolve toward awareness and intelligence, multiple databases and databases are emerging that must be properly interconnected in order to manage urban metadata. Urban metadata is a huge amount of dynamic and static data generated from various urban topics, organizations and even individuals. This data is collected by municipalities, government agencies, companies and individuals using new generation information technologies. Big data can be shared, integrated, analyzed and explored through the Internet of Things, cloud computing and artificial intelligence technologies to help people gain a deeper understanding of urban operations, as well as more informed decisions about urban management and A scientific approach led to the optimization of urban resource allocation, reduced the operating costs of the urban system, and at the same time to the safe, effective, coordinated and intelligent development of cities (Yunhe and Yun, 2016). Today, modern cities have evolved from dual spaces to triple spaces. The first dimension is space, which consists only of a physical

environment and all its resources are in a natural state. The second dimension includes the human community that is formed by the culture, norms and social interactions of the city's inhabitants. The third dimension, unlike the previous two, is a cyberspace that includes computers, Internet access, and data streams that are transmitted to information domains through various systems (Yunhe, 2015). Accessibility of data must be ensured in such a way that citizens can access the data system for free and can modify and correct this data. Citizens' involvement in this process can lead to more information angles as well as more data for citizens. The purpose of this article is to review the technologies and infrastructure architecture used in the smart city. In the following, we will examine the axes and infrastructure of the smart city, and at the end, the use of big data tools and techniques and the considerations of using big data methods will be examined.

2- The axes of the smart city

The European Union has divided the smart city into six axes (a set of features related to a field of work) based on various research studies (Saborido and Alba, 2020) (Figure 1).

2-1- Smart Economy (Competitiveness)

This axis includes factors related to economic competition such as population density analysis, digital marketing or tourism management. Information on the pattern and behavior of mobile users provides useful insights to enrich municipalities' decisions about tourism services and urban planning.

2-2-Intelligent environment (natural resources)

It describes natural conditions (climate, green space), pollution, resource management, as well as efforts to protect the environment, such as energy efficiency for buildings, intelligent lighting, or intelligent waste collection.

2-3- Smart governance (citizen participation)

This axis includes aspects of citizen participation in service delivery, and government performance.

2-4- Smart life (quality of life)

This axis includes various aspects of quality of life: culture (smart heritage), health (remote care via mobile phone), safety (flood detection) and housing (safe videos).

2-5- Intelligent mobility (transportation and communication technology)

This axis includes local and international access as well as the availability of modern / sustainable information and transportation systems.

2-6- Intelligent people (social and human capital)

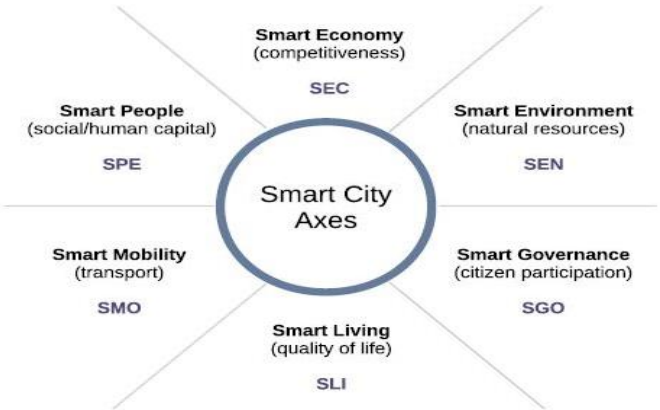

Figure 1- Smart city axes (Saborido and Alba, 2020

This axis is described by the quality of social life interactions and includes technological solutions for the development and implementation of teaching methods, providing a platform that assesses children's performance in each skill, individually and in groups.

In general, the exploitation of smart cities leads to the production of data with very rapid and exponential growth, as a result of which such volume of data or big data is the core of the services provided by the Internet of Things.

The big data phenomenon is characterized by the volume, speed and variety of types of data produced (Gani et al, 2016). Big data allows cities to gain valuable insights from a significant amount of data collected through multiple sources. Admittedly, such data generally include unstructured features (Chen et al, 2014). Figure 2 shows an overview of smart technologies with big data and cloud computing, in which various intelligent applications exchange information using embedded sensor devices and other devices equipped with cloud computing infrastructure to a large volume. Generate from unstructured data. This huge amount of unstructured data is collected and stored in data centers or in the cloud using distributed databases such as No SQL, which is shared between several services (Borgia, 2014). Therefore, a programming model with a parallel algorithm can be used to process large data sets to obtain valuable information from stored data.

Smart cities play a key role in transforming various areas of human life, such as transportation, health, energy and education. For example, weather data is growing very rapidly. Identifying and obtaining valuable information from a wealth of climate data can be very useful in terms of agricultural development. In addition, by analyzing climate data, people can be made aware of the possibility of dangerous events such as floods, extreme heat, drought, etc. (Fan and Bifet, 2016). The use of big data in smart cities can change any part of a nation's economy (Batty, 2013). These changes enable cities to implement the basics of learning and the practical needs of smart cities by understanding the basic characteristics of smart environments.

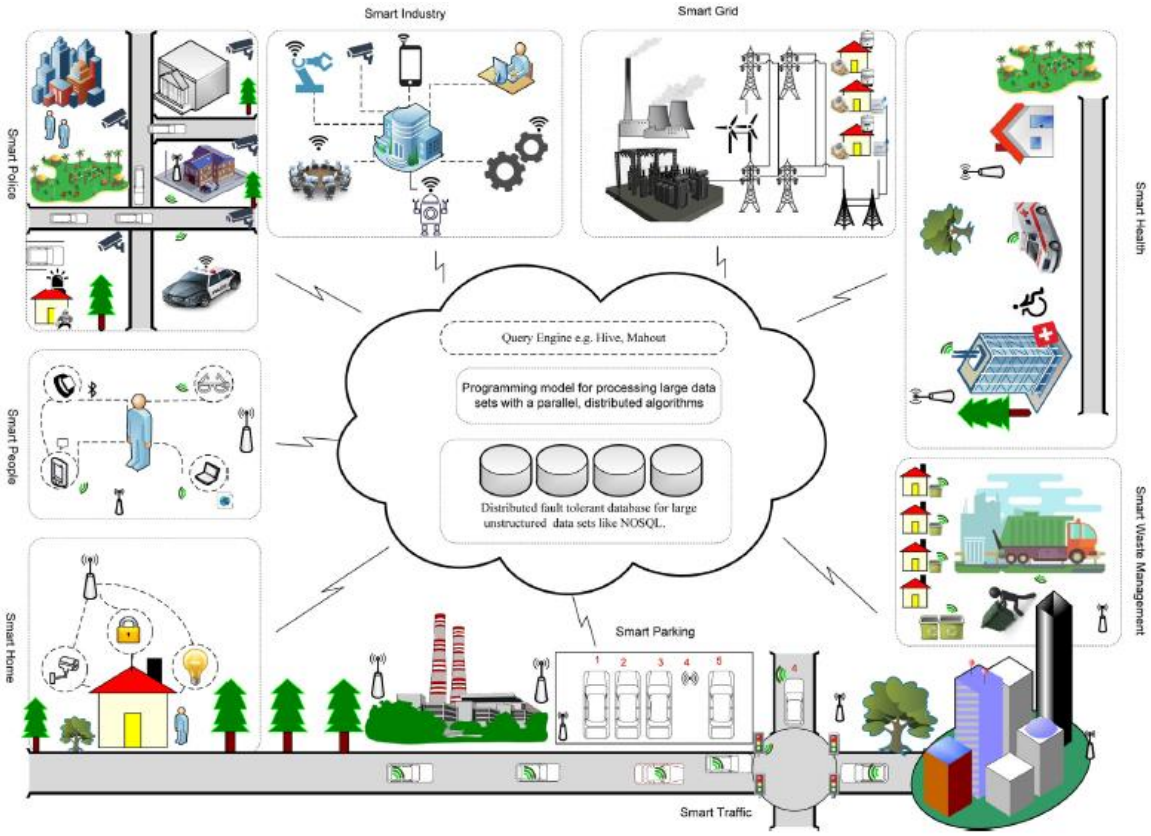

Figure 2 - View of smart technologies with big data (Targio et al, 2016)

3- Smart city infrastructure

In a smart city, data is as important as physical infrastructure. In the past, cities have developed precise mechanisms for managing physical infrastructure. For data, such a mechanism either does not exist or is in its infancy. Data infrastructure is required for data storage and sharing and has the following features (Neena, 2020):

• Reference - The data infrastructure is a credible source of data.
• Transparency - A transparent data infrastructure is where data comes from, how it is collected, and how it is processed.
• Openness - The data infrastructure is open to all users, making data as accessible as possible.
• Real time - the value of data decreases with age. Using the Internet of Things creates a huge amount of real-time data that can be accessed through the data infrastructure.
• Agility - In addition to the previous aspect, agility is in the rapid updating of data, the data infrastructure must be agile to accommodate new data sets.

Depending on security and privacy policies and data distribution policies in the city or country, data may be stored in cloud services or in large authorized data centers.

Figure 3 shows the infrastructure support model for urban data sets. The sensor layer detects and obtains urban data via the Internet of Things.

The network layer focuses on building integrated networks and information convergence. The data layer adapts a large

amount of data generated by the Internet of Things and information systems and thus produces a public urban database. The platform layer mainly includes various cloud computing services, public information platform and metadata analysis.

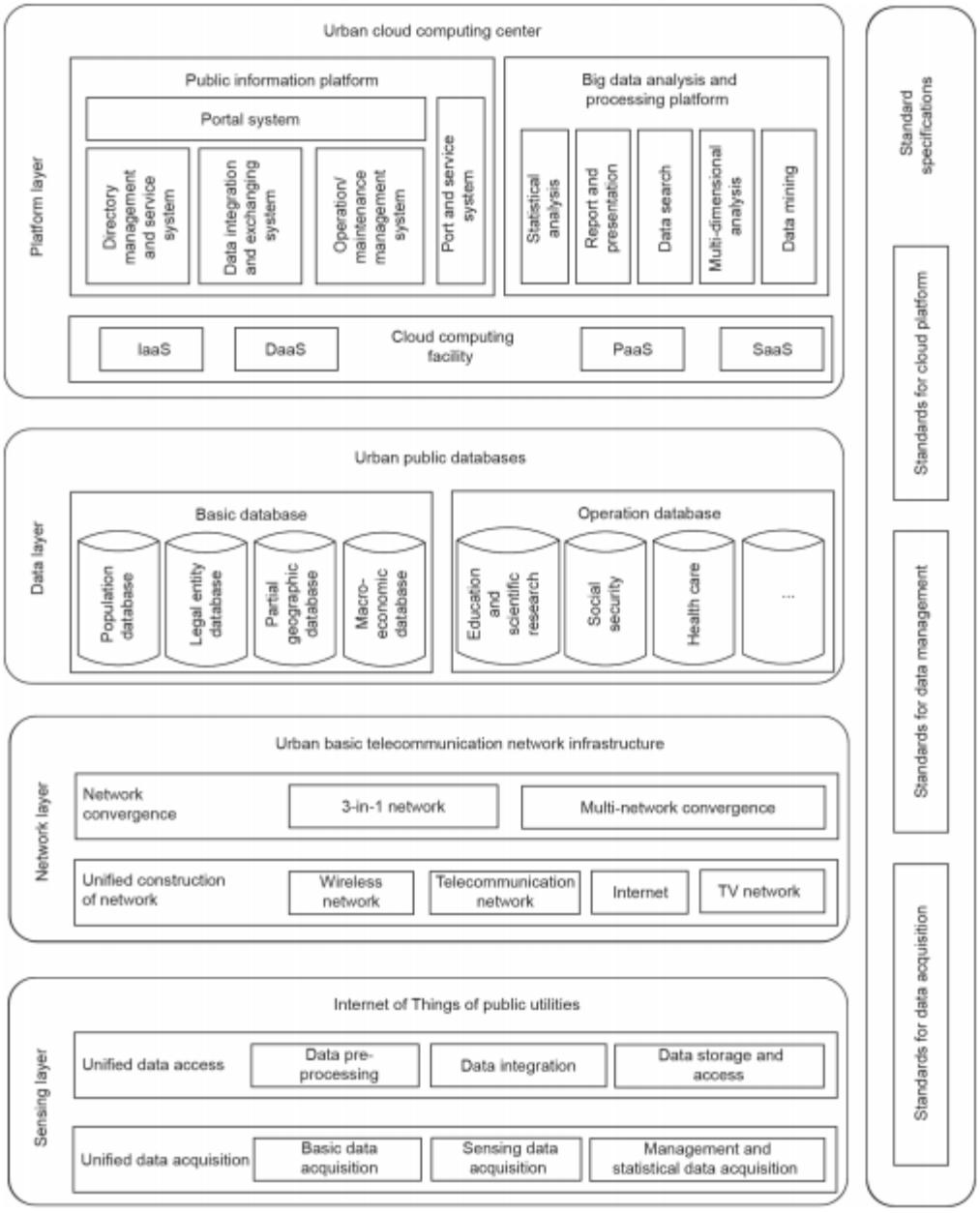

Figure 3. Urban data metropolitan infrastructure support model (Yunhe and Yun, 2016).

Another model of smart cities has a four-tier technology structure for capturing and processing data, as shown in Figure 4 (Neena, 2020):

1. Application layer
2. Application and data support layer and related processes
3. Communication layer
4- Information recording layer using sensors, CCTV cameras, citizens, news, social media and organizations and processes

The sensor layer may include real sensors placed by city officials to get information or even cell phone information from city residents, or other sensors placed on buses, motorcycles, bicycles, and so on.

These sensors may also record air quality or rainfall or water or gas leaks that occur anywhere in the city. Surveillance cameras may be deployed around the city to monitor traffic, accidents, or even street conditions or even garbage collection in the city. Intelligent processing in CCTV cameras can trigger emergency warnings for immediate or safety measures. Some crowded places or shopping malls and banks have started installing sensors that can detect the firing of bullets. Data from such critical sensors can help take immediate action in affected areas.

A network or communication layer connects data collected by various agents to the cloud space in a data center, which helps the data decision-making process.

Data can also be collected by satellite imagery or by a low-power network. In some smart grid applications that run throughout the smart city community, peak power consumption and even power outages can be managed by these networks as a precaution.

Programs and data support layers with processes can be country / city specific and can define data usage and storage rules. Applications are allowed to access this data based on the privacy and confidentiality of the data.

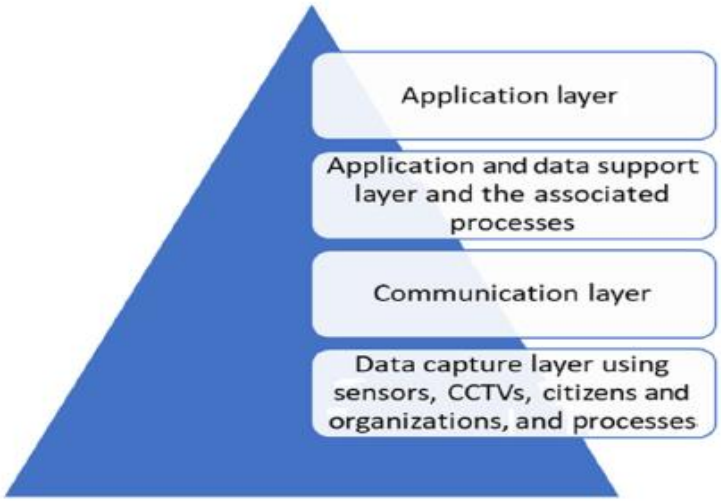

Figure 4- Smart city technology architecture (Neena, 2020)

In the smart city there is software to use on any device. Which enables them to communicate with other devices and the central database collection. This software system defines the smart city as an environment that supports software developers in designing,

implementing, deploying and managing applications for smart cities. . However, smart city software is different from software on desktops, tablets or even smartphones. Therefore, smart cities need to develop and use communication standards and protocols to allow heterogeneous devices to communicate and use common software. Many applications in smart cities are based on closed systems. They are usually private systems sold by a particular vendor, and do not provide much detail about the design and architecture of the software. While open systems pursue cooperation on platforms and mobility without software lock. However, both closed and open software systems can provide public APIs, also known as open APIs. Communication protocols compatible with these systems usually differ in their interaction models, which are: Publish-Subscribe and Request-Response.

The diffusion-sharing model provides a distributed, asynchronous, and interconnected relationship between data producers and destinations. The request-response communication model allows the client to receive information from the server that receives the request message, then processes it and returns the response. Despite the various communication protocols in the Internet of Things such as AMQP, CoAP, DDS, MQTT, REST HTTP, XMPP, two perfectly suitable options that are also of interest to developers. The MQTT and REST protocols are HTTP. MQTT is an IoT connection protocol designed as a lightweight publishing / subscription messenger. REST HTTP is based on the request-response model and emphasizes component scalability, independent deployment of components and reduction of communication latency, security implementation and encapsulation (Dizdarevic et al, 2019).

4- Tools for collecting and analyzing big data

In order for big data to reach its main goals and advanced services in smart cities, it is necessary to use the right tools and methods for efficient data analysis. These tools and methods may strengthen cooperation and communication between existing entities. As shown in Figure 5, the macro data system structure in the smart city can be divided into several layers to implement integrated macro data management and smart city technologies. Each layer represents the potential performance of the smart city components. The first layer is a set of objects and devices that are connected to each other through local area networks and / or wide area networks. Most of these objects and devices actively generate large amounts of unstructured data every second. In the second layer, all data collected without structure is stored in a common distribution database, which can be in the city data center (which is equipped with all network elements) or by a large data warehouse such as S3, Google and Azure cloud services. In such cases, various big data storage systems such as Cassandra, Hbase (George, 2011), MongDB, CouchDB, Voldemort, DynamoDB and Redis can be used. In the same layer, the stored data is related to the queries received using a batch programming model such as the MapReduce framework or other big data processing engines. MapReduce is a powerful programming model for parallel and distributed mass data processing. In streamlined processing, data must be processed quickly so that companies and individuals can react immediately to changes in the smart city environment. Many technologies, such as Spark, Storm, and S4, can help streamline real-time, unstructured data.

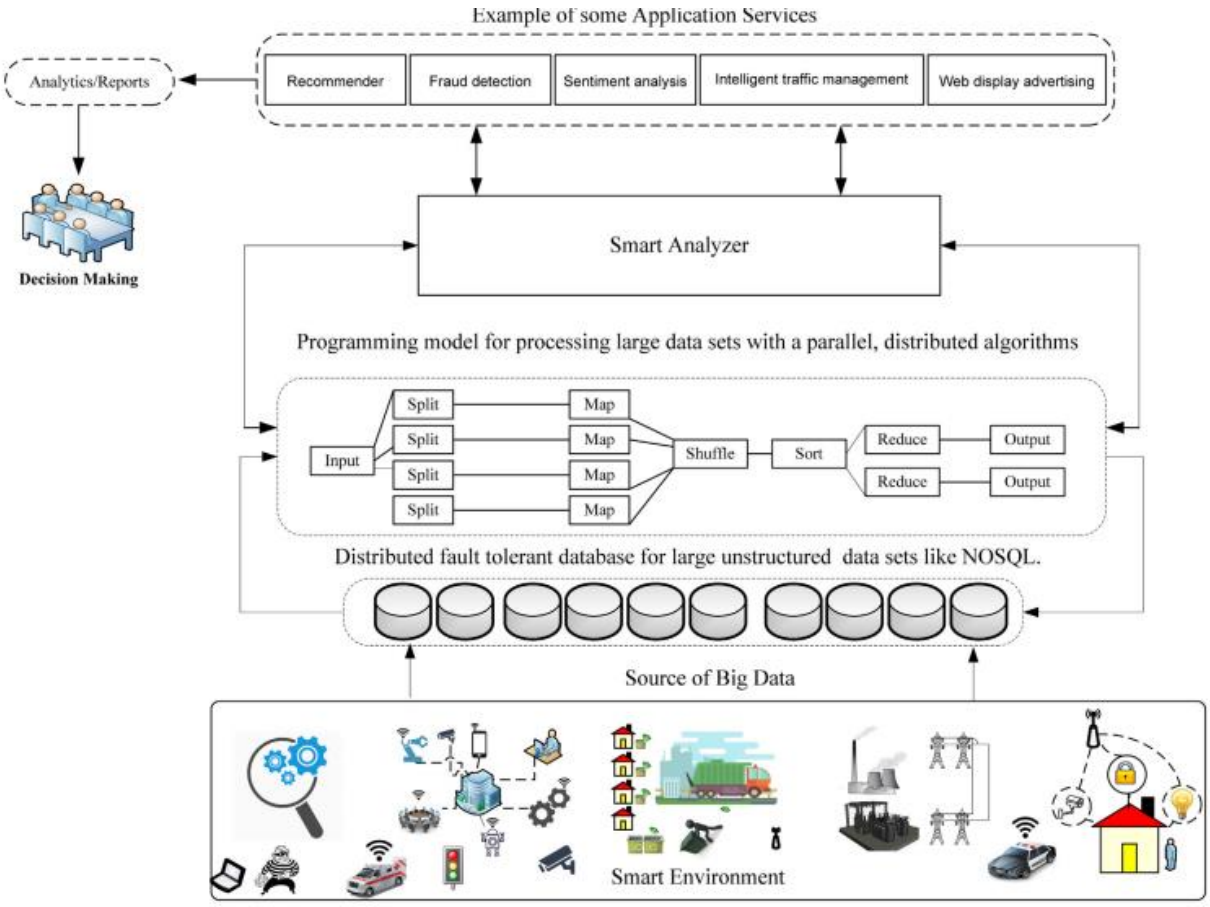

Figure 5 - The macro data system structure in the smart city (Targio et al, 2016)

Mahout is an example of this technology in which there are several machine learning libraries for filtering, clustering and categorizing data. The last layer is application services where people and machines communicate directly with each other to make smart decisions. These applications can be used for a variety of purposes such as fraud detection, emotion analysis, intelligent traffic management and web display analysis.

4-1- Open source tools for big data

Tools and frameworks address the issue of big data processing in two ways. Data parallelization, in which data is split into controllable pieces and each subset is processed simultaneously, or task parallelism, in which the algorithm is divided into steps that can be run simultaneously on the same data set. Run. Many big data solutions use complex workflows and require systems built using a combination of advanced tools and techniques. The most prominent and widely used open source tools in big data analysis are the Hadoop ecosystem projects (Umit et al, 2020).

4-1-1- Hadoop ecosystem

As data grows, the most challenging issue is scalability in architecture. Infrastructure scalability addresses the changing needs of a big data analytics program by adding or removing resources. In most cases, this is achieved by vertical scaling by adding more resources to the existing system to achieve an

optimal performance state or horizontal scaling by the distributed architecture.
Apache Hadoop is an open source platform for processing large datasets that uses a horizontal scalability architecture instead of vertical scalability. In addition, it provides error tolerance through the software and controls server failures. This means that cost-effective servers can be used for scaling first, rather than expensive, error-resistant servers. Second, both batch or stream processing tasks as well as data extraction, loading and conversion operations can be performed. Third, instead of moving data between clusters, code transfer is more efficient and faster for data processing, which is one of the advantages of Hadoop. Finally, because application development for a computer is simpler and more manageable than distributed application development, Hadoop provides a simple, controllable code framework that frees developers or analysts from the complexities of writing code for Keeps distributed systems away.
The Hadoop ecosystem consists of four modules:

Hadoop Distributed File System: The file system that is located at the bottom of the Hadoop architecture and consists of names and data by holding copies of nodes together.
MapReduce: A two-part data processing engine, mapping raw data to key pairs - reducing the amount and processing operations by combining and summarizing the results in parallel.
• Yarn: A resource manager that allows you to differentiate between infrastructure and programming models.
• Common: A set of common programs such as compression codes, input / output programs and error detection.
The overall structure of the Hadoop ecosystem consists of three layers: storage, processing, and management.

4-1-2- Storage layer

This layer includes the distributed file system and provides a distributed architecture for storing information. This file system consists of two main components NameNode and DataNode, which basically work with the master and slave architecture of the server and receiver. The NameNode is the core of the distributed file system and is responsible for disseminating information to other nodes that maintain and control block metadata such as the name, size, and number of blocks. Due to the unique nature of responsibility, NameNode is a vulnerability. DataNodes represent devices used to store system data blocks (Syed, 2018). By default, each data block is copied to three different DataNodes for availability and fault tolerance. NameNode is also responsible for monitoring the health of DataNodes using a method called heart rate. DataNode sends the heartbeat at predetermined intervals to let it know it is alive. NameNode also acts as a load balancer. The storage layer not only consists of a distributed file system but can also contain non-relational databases. These databases are also located in the storage layer of the Hadoop ecosystem. Data can be retrieved from the Hadoop file system using some open source tools such as Sqoop query languages (it feels like SQL, but not exactly the query language). These databases support nested, semi-structured, and unstructured data that we typically encounter with rapid data growth.

Some storage layer tools:

Sqoop: A tool used to exchange data between relational databases and the Hadoop ecosystem. And it is useful when using a distributed file system as a commercial data warehouse preprocessing engine.
Hive: The storage layer also includes data integration tools such as Hive, which allow standard SQL-based searches using HiveQL on data stored in distributed file system and

NoSQL databases. This is a powerful and simple way to query the system. Metadata of tables and partitions are stored in Hive Metastore. HIVE provides an interactive way to work with big data in a cluster and an easier way to write MapReduce code in Java. Which is very optimal and scalable. Hive is suitable for processing online transactions and stores data as text files. Due to the lack of an update database, inserting or deleting at the record level is not allowed.

4-1-3-Processing layer

Macro data analysis is obtained at the processing layer. Yaren creates an environment in which one or more processing engines can work on the Hadoop data cluster at the same time. . These two components were skilled in resource management but created bottlenecks so they were inefficient. To overcome these bottlenecks, Hadoop has introduced Yarn instead of tracker models in version 2. Yarn divides tracker performance into four services (Umit et al, 2020): Resource Manager, Node Manager, Container Manager, and Program Manager. The processing models used by data processing engines are classified as batch or stream. Batch processing is used in large datasets, and when completed, the output is written to a file or database. In batch processing, nothing is done in real time, while in stream processing, the data is processed when it reaches the system. This is good for real-time data analysis, but this type of processing requires special techniques to ensure that results are achieved in a meaningful time. The following is a list of keywords used to evaluate processing engines.

Delay: The time between starting work and initial results.

• Operational capacity: the amount of work done in a specific period of time, efficiency.

• Fault tolerance: Failure detection and recovery options.

• Usability of your engine: complexity of installation and configuration, interface language and programming.

• Resource cost: financial and time cost.

Scalability: This indicates the system's ability to adapt to incremental requests. The scalability of a processing engine is a key factor in trying to answer whether there is a bottleneck as the input or cluster size grows.

• MapReduce

Using ideas from functional programming and Google Library, Hadoop introduced the MapReduce framework for ease of parallel processing. The MapReduce model consists of two main operations, Map and Reduce. Map is the process of retrieving data from storage space, using algorithms, and generating results as key-value pairs. Reduce is the process of performing aggregate functions such as addition, multiplication, on key-value pairs. Some operations, such as mixing and sorting, can be performed to process data between Map and Reduce.

Similar to the distributed file system, MapReduce also has a master node that regulates the operation of the entire cluster and other nodes. Whenever a task fails, the fault tolerance process is done only by re-executing that task. If the processing speed of one node is slower than other nodes, that node is called unused and the same tasks are assigned to other nodes.

Map Reduce is also used for machine learning, in which the training dataset is read in its entirety to build a learning model. In a batch workflow, data is read from the file system distributed to the mapper. The Mapper generates those value-key pairs for sorting on disk and writes that after sorting, some data is sent to Reduce to teach the model. This read

and write operation can be inefficient in terms of time and computational resources.

• Spark

Apache Spark is a cluster computing environment that uses ideas similar to the MapReduce model, but improves speed by using computations within memory. Its response time is significantly faster than MapReduce in processing stored memory tasks and Hadoop operations on disk. Stores data in memory and provides RDD fault tolerance without duplication, using an abstract concept called flexible distributed data sets. RDD can be understood as read-only distributed shared memory. RDD was generalized to include Data Frames. This allows a group of data to be collected by the columns, so RDD can be thought of as a schema. The learning process takes place through the average cache of the results. Spark easily supports integration with Java, Python, Scala and R programming languages. Supports multiple data sources including Cassandra, HBase or any Hadoop data source. In addition to effective features, Spark has some inefficiencies in streamline processing, and bottlenecks may occur due to data transmission through nodes using the network (Al-Turjman, 2018).

Spark has various processing components that are simply as follows:

Spark core: It is the main component of Spark and important functions such as work scheduling are performed here.

1- Spark SQL: is a structured data processing unit that executes SQL and Hive query languages for structured data.

2. MLib: A machine learning library for implementing algorithms such as classification, regression, decision trees, and random forests.

3- GraphX: Designed for graph analysis.

• Storm

Apache Storm is a distributed processing framework for processing flow data written in the clojure programming language. Storm performs stream processing in real time. And consists of input current and calculation logic. Computational logic can process data from any input stream. Storm was originally implemented in Java, but is now used to develop various languages. Error tolerance is obtained by tracking called the main node. If Nimbus has a problem, it launches automatically with a different tracking method from MapReduce. Storm also supports the "Lambda" architecture, which is a way to break processing into three layers: batch, service, and speed. MapReduce and Storm can perform tasks simultaneously that can be used to process both real-time and historical data. Storm does not include a machine learning library, but the macro data mining platform provides implementations for routine classification and clustering algorithms.

• Flynck

Apache Flynck is a framework for processing unlimited flow data in clustered environments. Infinite data can be described as an ever-growing, and essentially infinite, set of data. This is often the case with streaming data. Limited data are like batch data sets (Umit et al, 2020). Flynk has the ability to process batch and stream data in real time. Flynck can be integrated with both the distributed file system and Yarn, as well as running independently of the Hadoop ecosystem. A machine learning library called Flink ML was also introduced. Other machine learning libraries, such as the macro data mining platform, can also be used.

• H2O

H2O differs from other processing engines in its web-based user interface. This interface makes machine learning more understandable and accessible to non-technical users. The library also offers parallel processing engine, analysis, math and machine learning along with data preprocessing and evaluation tools. The framework supports Java, R, Python and Scala programming environments.

4-1-4- Management layer

The management layer is responsible for scheduling, monitoring and coordination and provides the user interface. As storage and processing tools are used, task creation requires careful organization and regulation. This high-level configuration and user interaction takes place at the management layer.

Some of the tools in this layer are as follows:

• Oozie

It is a system for executing and planning Hadoop tasks. It is essentially an interactive workflow planner that allows things to be chained together. For example, MapReduce, Hive, Pig, Sqoop, and more can be chained together. Workflow as a rotation chart marked in xml. Therefore, it is possible to perform an action that is not dependent on another.

• Zookeeper

It is a service for synchronization and synchronization and tracks information about the main nodes and other nodes. It is a tool that applications can use to recover partial failure. Can manage partition and clock breakdowns. The program supports Java and C and can also communicate with Perl, Python clients. The main operations of this service are: main selection, loss detection (error) and group management.

• Hue

Hue provides a web interface for the Hadoop ecosystem. It has a file browser for the distributed file system and a working browser for MapReduce and Yarn. Hue can be used to manage interactions with Hive, Pig, Sqoop, Zookeeper and Oozie, and also provides tools for data visualization.

5. Considerations in using data for smart cities

In the previous sections, the tools, techniques, and infrastructure needed to expand big data to become a data-driven smart city were described. However, as with any large-scale change, moving to a data-driven smart city is not an easy task. Considerations that depend on the use of metropolitan data in the implementation of smart city projects should be determined. In this section, we focus on these considerations (Lima et al, 2018).

5-1- Service orientation in viewing and using data

The first is to adhere to a service-oriented approach to creating value in the collection and analysis of urban data. There are many paths to using a data set, and all the distinct sections such as data collection, integration, management, and data analysis are organized according to different paths. Therefore, a prerequisite for using metadata for a smart city is to define the directions or themes of the applications that are developed. The final value of the data) can be significantly increased.

5-2- Customer experience about data and information

The second consideration is to pay attention to the experience of individuals (eg citizens) in collecting and using personal data and providing information to them. Creating interesting experiences and instant moments is essential for customers of any kind of service.

Therefore, attention to personal experiences (for example, the experience of citizens) is also essential for the context of the smart city. In particular, it is important that citizens do not complain about privacy issues (for example, data from places they have driven) (for example, that they do not have a problem with the conditions or timing of the data recording). Not only should the information provided to customers be useful, but the method of data collection should provide a natural user interface for data collection, privacy should be taken into account in the use of data, and the visualization of the information content should be clear and rich. And the provision of end-to-end services should generally be for customers.

5-3- Data orientation in designing service applications

The third point is to use a data-oriented perspective in service design. Smart city projects must design credible and workable services for public purposes, and must consider the technological aspects of the services. Therefore, understanding the methods of collecting, managing and analyzing information to design credible services is key. And the design of data-driven government services should be distinguished from other public services. The data-oriented perspective is like the method of data validation and the feasibility of data analysis and must be applied to obtain valid services, because the data in question is the main source of service creation and value creation for local communities.

5-4- Cooperation and interaction between data-related stakeholders

The fourth consideration is to create coherence and minimal conflict between data related to stakeholders. Stakeholders are people who participate in the use of municipal services. Creating value in using big data in smart cities involves different types of stakeholders. The art of using big data for smart cities is to try to address the concerns of different stakeholders. (For example, in providing citizens' health services, no health services should be provided that may be inaccurate or controversial, and the scope of work of individual physicians should never be exceeded.

5-5- Existence of different views about data and applications

The fifth consideration is the formation of a multitasking group to use big data. While considerations include one to four psychological challenges in using big data for smart cities, these considerations address cultural and organizational challenges. Naturally, any urban improvement project involves soft tasks that require multidisciplinary human activities. Using big data to advance public services requires organizing a team with members with different specialties from different operational units, including planning, design, engineering, information technology, statistics, and management, which in fact this group should include different types of experts. Such as transportation experts and physicians, data researchers, IT professionals, business experts and government employees, in this regard another artistic aspect of using big data for smart cities is integrating the expertise of different professionals into a body of knowledge to analyze Data analysis and service design. It is necessary to form a group of experts from different fields and specialties to solve the challenges. By professionally integrating people with multiple backgrounds, ideas for addressing challenges and considerations can be identified.

6-Conclusion

Proper use, management and utilization of urban metadata technologies promotes the development of urban knowledge-based services, so today's society should prioritize

the design and development of platforms for big data processing. In this paper, smart city axes and big data tools and techniques along with infrastructure models in implementing smart cities were examined. In the end, we came to the conclusion that using existing technologies in big data to discover effective patterns will be an integral part of managing today's developing and smart cities. However, current tools are still in the early stages of deployment, and providing more appropriate services in some areas, including the development of new powerful analysis techniques and tools, and distributed and scalable exploration methods, should be considered by researchers.

7- Resources

1. Al-Turjman, F.Fog-based caching in software-defined information-centric networks.Computers and Electrical Engineering, (2018), 54–67.
2. Batty, M. Big data, smart cities and city planning. (2013). Dialogues in HumanGeography, 274–279.
3. Borgia, E. ”The internet of things vision: key features, applications and openissues .Computer Communications, (2014) 54, 1-31.
4. Chen, M., Mao, S., & Liu, Y. Big data: a survey. Mobile Networks andApplications, (2014) 171–209.
5. Dizdarevic, J., Carpio, F., Jukan, A., & Masip-Bruin, (2019). X. A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. ACM Comput. Surv 51.116: 1–116: 29.
6. Fan, W., & Bifet, A. Mining big data: current status, and forecast to thefuture. (2013). ACM SIGKDD Explorations Newsletter, 1–5.
7. Gani, A., Siddiqa, A., Shamshirband, S., & Hanum, (2016) F. A survey on indexingtechniques for big data: taxonomy and performance evaluation. Knowledge and Information Systems, 46 (2), 241–284.
8. George, L (2011). HBase: the definitive guide. O’Reilly Media Inc.
9. Lima, Chiehyeo. Kimb, Kwang-Jae. Maglioc Paul (2018), Smart cities with big data: Reference models, challenges, and consideratio, Cities, pp. 86-99.
10. Neena Pahuja. (2020). Smart cities and infrastructure standardization requirements. Solving Urban Infrastructure Problems Using Smart City Technologies. Pages 331-357.
11. Saborido, Ruben. Alba, Enrique (2020). Software systems from smart city vendors, Cities
12. Syed, M. F. A. (2018) Big data architect’s handbook ”. Birmingham, UK: Packt.
13. Targio, Hashem. Ibrahim, Abaker. Chang, Victor. (2016) The role of big data in smart city ”International Journal of Information Management, Volume 36, Issue 5, October, pp. 748-758.
14. Umit Deniz Ulusar, Deniz Gul Ozcan, Fadi Al-Turjman. (2020). Open Source Tools for Machine Learning with Big Data in Smart Cities Smart Cities Performability, Cognition, & Security, 153-168.
15. Yunhe Pan, Yun, Tian. (2016). Urban Big Data and the Development of City Intelligence, Engineering, Volume 2, Issue 2, June, pp.171-178.
16. Yunhe, Pan. (2015). Project Group of Strategic Research on Construction and Promotion of China’s iCity. Strategic research on construction and promotion of China’s iCity. Hangzhou: Zhejiang University Press.