International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-11 Issue-5, January 2023

Overview of Big Data Analytics Technologies in Smart Grid

Aditya Arya, Sridhar S

Abstract*: Smart grids have become an essential component of modern society due to their interconnected nature. In the smart grid, unprecedented amounts of data will be created continuously due to the advanced sensor infrastructure. Therefore, analyzing smart grid data is becoming increasingly critical to delivering electricity and managing consumption in the business and physical sectors. Modernizing the grid requires data science, despite the challenges of integrating data analytics into the enterprise. A review of big data management & analysis in the smart grid is presented in this paper. Data analytics and its role in big data management are discussed in this paper along with the challenges of implementing those analytics, and how they can help achieve clean, reliable, and efficient grids. The paper supports Apache Flink due to native streaming for use cases that call for minimal latency, while Apache Spark is better suited for batch data processing.

Keywords: Smart Grid, Big Data, Intelligent Grid, Grid Analytics.

I. INTRODUCTION

Intelligence is embedded into all electrical grids and components to manage them more efficiently [1]. As a result of its size and complexity, grid systems include many devices and assets connected to a network that can be controlled, monitored, tracked, and managed as needed. The big data analysis of smart grids can provide new solutions based on the measurements available, enabling consumers to meet their demands, forecasting new loads, and creating new ideas and perceptions. Another related topic is measuring consumption data from smart meters and faults or defects encountered. So, big data analytics can provide solutions to the topics mentioned above. Big data are substantial data sets that cannot be captured, managed, and processed with traditional techniques and tools. Various data sources, such as advanced measurement infrastructures (AMIs), scattered distributed generators, and phase measurement units (PMUs), can generate big data. A PMU, for example, can generate 50 or 60 phase measurements per second. Energy requirements are increasing rapidly especially in countries like India. India's power demand increased from 106.6 billion units (BU) in 2019 to 124.2 BU in 2021 and 132 BU in 2022. Data analytics and predictive analytics will fundamentally change the electricity industry. Making the grid more efficient will be easier with the integration of Big Data technologies. Moreover, it will fundamentally affect how electricity is priced, who sells electric power, and how utilities, grid operators, and consumers interact [2-3].

One must first take a step back and understand how the electric power industry differs drastically from most other industries to comprehend software and hardware's impact fully. Utilities typically do not compete over market share or increase revenue and margins. In contrast, they enjoy legal monopolies, stable pricing, and predictable recovery of investments in fixed assets. Therefore, it is more important to them to maintain quality, stability, and reliability than to generate revenue. Additionally, these institutions are public or quasi-public with fiduciary responsibilities and provide essential services to all within a service territory, improving the general public's well-being and safety considerably [4-6].

Utility companies are starting to use smart meters in place of the outdated ones that require human readings once a month. Smart meters give updates on usage on a periodic basis (every 15 minutes). Utility providers are experimenting with taking readings every 30 seconds in various situations. The switch to smart meters generates a significant volume of streaming data and has significant potential advantages [5]. Using machine learning models to identify usage abnormalities brought on by equipment malfunctions or energy theft is one of the benefits. These new objectives cannot be achieved without effective means to transport and reliably process streaming data at high throughput and with very low latency [6].

II. WHAT IS BIG DATA

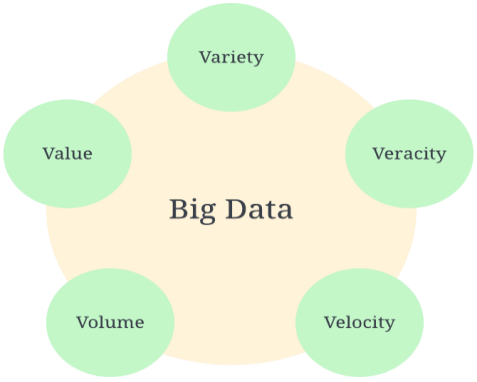

Fig. 1: Five Vs of Big Data [1]

Manuscript received on 02 December 2022 | Revised Manuscript received on 13 December 2022 | Manuscript Accepted on 15 January 2023 | Manuscript published on 30 January 2023.

***** Correspondence Author (s)

Aditya Arya*, Department of Electrical and Electronics Engineering, MS Ramaiah Institute of Technology, Bangalore (Karnataka), India. Email: adityaarya3510104@gmail.com, ORCID ID: https://orcid.org/0000-0002-2464-7656

Dr. Sridhar S, Department of Electrical and Electronics Engineering, MS Ramaiah Institute of Technology, Bangalore (Karnataka), India. Email: sridahrs@msrit.edu, ORCID ID: https://orcid.org/0000-0002-9127-9208

Amount of data that cannot be analyzed by traditional tools is known as big data. Big data is generally defined as 5 V's: Volume, Veracity, Variety, Velocity, Value.

- Volume: Big data itself means enormous data. Smart grid generates a vast amount of data that cannot be stored or analyzed using traditional database technology.
- Velocity: Refers to the speed of data generated by different data sources. In order to make the best business decisions, businesses need their data to flow quickly so it can be analyzed and the business decisions can be taken accordingly.
- Veracity: Veracity refers to inconsistencies and uncertainty in the data, which affects the big data analysis. Multiple sources produce a multitude of data of different data types.
- Variety: Data is collected from different sources. Thus, collected data can be structured, semi-structured, or unstructured. An unstructured data set is unorganized and can come in various file formats; unstructured data is not easily incorporated into any conventional data model. The term semi-structured data refers to data that is not organized into a specialized repository but has metadata associated with it. This allows it to be processed more efficiently than unstructured data. On the other hand, a structured data set is a set of data whose schema or model has already been defined. By doing so, the data can be processed and analyzed more effectively.
- Value: Refers to the value companies derive after analyzing the data—the value of big data increases based on the insights it provides.

III. SMART GRID SYSTEM

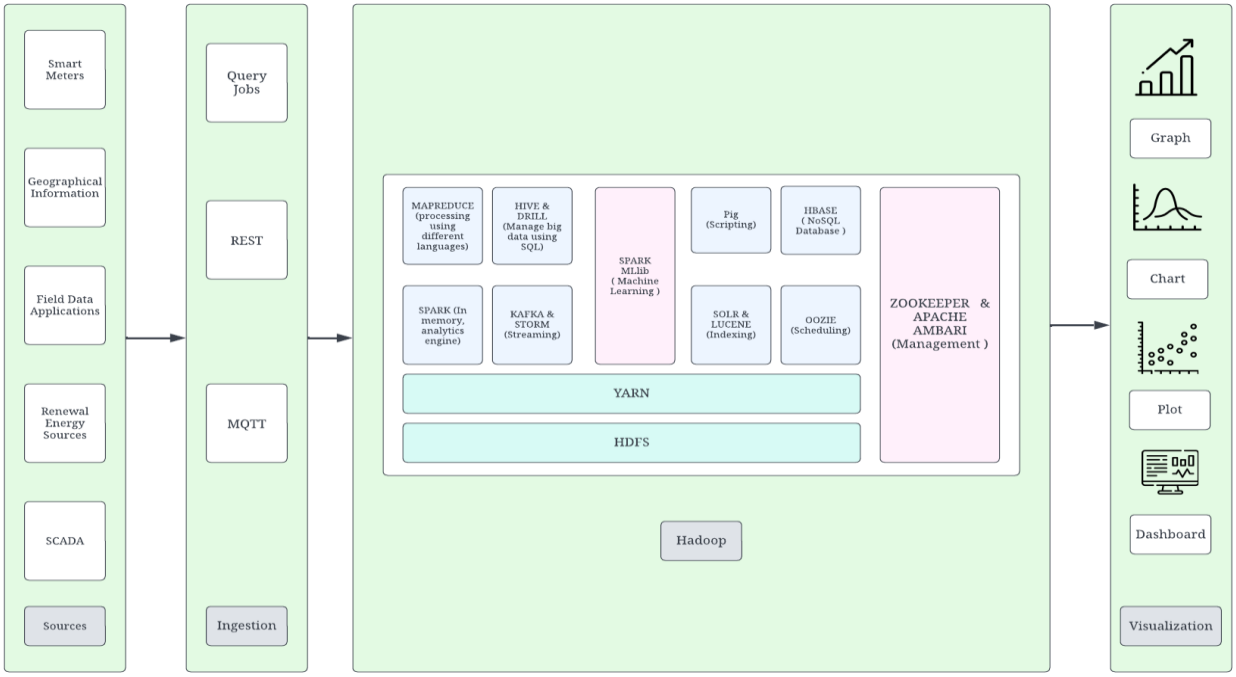

Fig. 2: Big Data Pipeline [6]

In figure 2, a big data pipeline diagram is presented. The big data pipeline starts as data is produced by the sources (IoT devices, Electric Machines) and is then ingested using various tools such as Kafka streams, REST Api, and MQTT. Data ingested is now stored in a distributed system like Hadoop, which comprises a range of tools that can be used to analyze the data. In the end, data is visualized to extract meaningful incites from it [6].

1) Data generation, storage, and management

In the smart grid, computer technology is used to connect, automate, and improve communication among the various components of the power network. As a result, consumers and transformers can share information more easily. One key element of this system is installing smart meters in homes and businesses. Smart Meters installed in homes, businesses, and industries are the primary source of data generation in smart grids. Using digital meters, producers and consumers can communicate and exchange supply and demand data [7].

The operation of the smart grid depends on the data gathered by smart meters. Power producing facilities, for instance, can make better predictions and management of peak demand using this data.

As a result, they are able to scale back production when less energy is required and quickly ramp it up when demand is at its highest. The smart grid employs computer, communication, and data processing technologies to increase flexibility and efficiency. Electric vehicles and other new network strains are also introduced, while opportunities are created for intermittent generation technologies like solar and wind. A smart grid generates vast amounts of data, which poses new challenges for data storage and management. Big data in the context of the smart grid is typically unstructured, has large dimensions, and is produced frequently in real time.

Traditional database management systems cannot properly store massive data in the smart grid because of these qualities. Sensors that transmit smart grid data are widely distributed, and it does not make sense to store them in a centralized manner; storing them in distributed systems is more practical. Third parties can easily access this data to provide different analyses. However, sharing data has its own set of challenges as data needs to be shared securely. Therefore, sharing data securely with different vendors would be a challenge [8].

2) Data Sharing

In the smart grid, data is shared and transported across numerous stakeholders at various levels. Big data is initially moved from sensor infrastructure to digital infrastructure (databases, data warehouses, data lakes). Then, at the higher level, different parties can share critical private data collaboratively to achieve a specific goal. Therefore, it is paramount to ensure data security at either level. Here, data security has two-fold meanings.

In the first case, the data is confidential, meaning it cannot be disclosed to the outside world during communication; in the second case, the data is consistent, and integrity is protected against outside manipulation. However, there may be severe consequences if data is not communicated securely. Cyber attackers can induce inaccurate control decisions by fabricating data. The disclosure of confidential bid information might also result in the generation company losing profits.

3) Data Analysis

At this step, we have data now we have to make sense of it. Data analysis is crucial for understanding business problems and exploring data meaningfully. In this case, data findings are recorded, analyzed, and presented in an easy-to-understand manner for the utilities to interpret and make decisions. Data analysis is essential if a business wants to run smoothly and effectively. With a sound data analysis system, utilities can determine the sectors of their business that need more investments. In addition to assisting the company in making informed decisions about its business, data analysis could assist the company in avoiding losses [9]. The goal of this paper is to explore some emerging technologies and some potential applications for big data in smart grids.

IV. DATA IN GRID

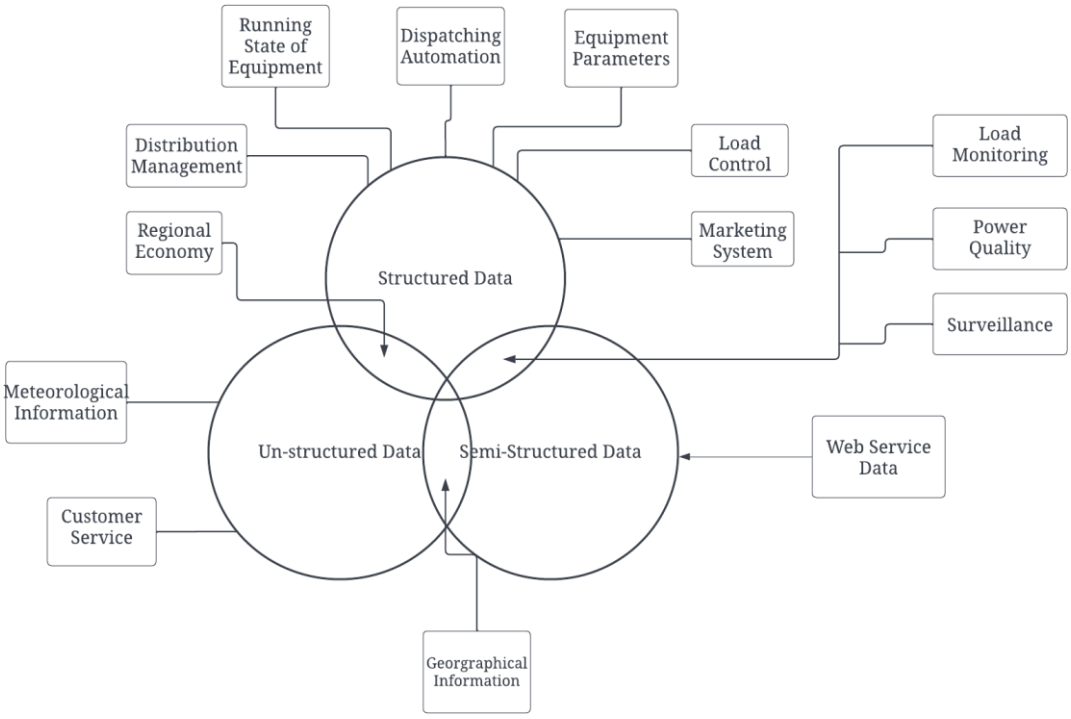

Fig. 3: Different data structure and their sources [9]

Data means information that needs to be stored in the correct format to be processed and give meaningful insights. Intelligent systems such as smart grids produce data from the process of electricity generation, transmission, distribution, and consumption. In figure 3, different data structures found in smart grid is presented. These data include information from distribution stations, electricity meters, and nonelectrical sources, which may be used to improve energy production planning and marketing strategies, as well as to provide accurate information on the use of energy in a particular region [10].

A. Structured Data

In structured data, the model is well-defined, the structure is well-defined, the order is consistent, and the data is easy to read and understand by a user or a computer program. Structured data can be stored in spreadsheets, CSV files, relational databases, online analytical processing, data cubes,

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

relational tables containing customer information, and electrical consumption data. In smart grids, structured data includes information from meters, distribution management systems, equipment parameters, load controllers, and marketing systems [10].

B. Semi-Structured Data

Textual information that can be parsed and has observable data patterns is referred to as semi-structured data. Data lacks a relational database but possesses some organizational characteristics that facilitate analysis. JSON and XML files, for instance, are self-descriptive. Semi-structured data in smart grids includes data from load monitoring, web services, and power quality.

C. Unstructured Data

As unstructured data lack's structure and cannot be identified easily by computer programs, it cannot be used easily. Unstructured data can be collected publicly through census, text, social media streams, and tweets.

V. SMART GRID DATA FLOW

An intelligent grid consists of automation, electrical networks, and information and communications technology. Electrical networks in the smart grid need smart meters, sensors, and control devices. The development of smart grid technologies has coincided with the growth in accessibility and environmental friendliness of renewable energy sources. To fulfil baseline, moderate, and peak loads, the smart grid must utilize all available energy sources. Big data analytics has a number of benefits for the smart grid beyond only generating intelligence from raw data and extracting information from it [13-14]. The figure 4. shows the breadth of big data analytics in the smart grid. For big data architectures to be successful, it must be possible to perform a variety of analytics on large volumes of data to generate useful business applications.

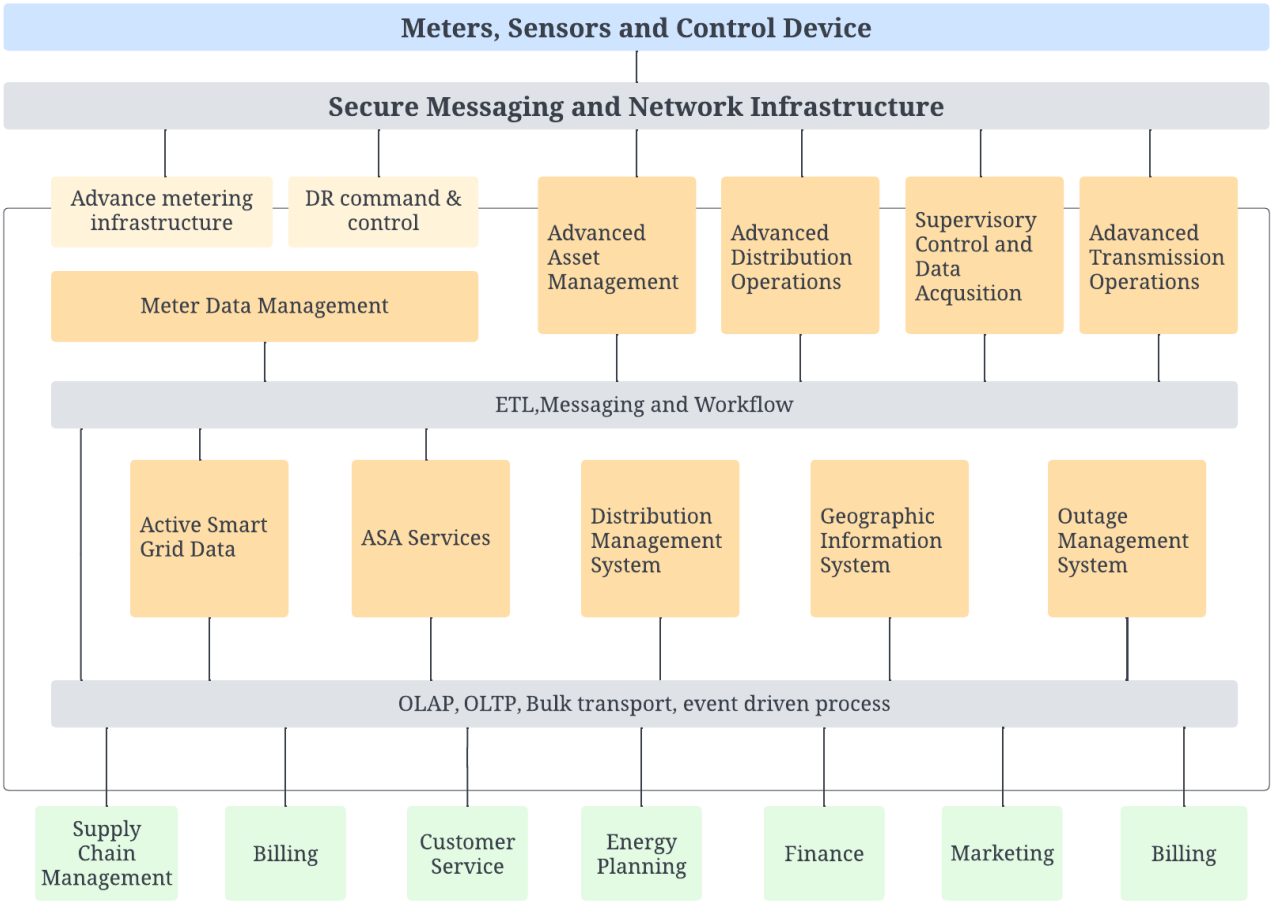

Fig. 4: Smart Grid Data Flow [11]

VI. BIG DATA ANALYTICS PROCESS

Process of finding patterns, relationships, and trends in massive datasets that cannot be discovered with traditional data management techniques and tools. Big data analytics cannot be done using a single tool or technology. Instead, several tools work together to help collect, cleanse, analyze, and visualize big data [12].

Big data analytics requires a predefined strategy as vast amount of data is required to be processed and an infra check can help analyze the data in more meaningful manner. Some of the components that involves modern data analytics consists of Flume, Apache Hadoop, Apache Spark and data visualization platform such as Power BI or Tableau.

Types of Big Data Analytics:

Retrieval Number: 100.1/ijrte.E73880111523
DOI: 10.35940/ijrte.E7388.0111523
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

- Descriptive Analytics: In this, historical available data is analyzed, and represented in form graphs, bar diagrams, pie diagrams, maps, scatter plots. This step is performed to find pattern in data.
- Predictive Analytics: This form of analytics uses different statistical and mathematical model to predict future events. In terms of smart grid, this can be used to predict load requirements.
- Exploratory Analysis: There is often correlation between data, this step helps us to find hidden pattern in the data, which can be further analyzed to find cause of an event.
- Prescriptive Analytics: This is used to determine the optimum result for a particular condition. Prescriptive analytics is essentially used to figure out how to influence future events to be more favorable [15-16].

VII. BIG DATA TECHNOLOGIES

A. Map Reduce

MapReduce is a big data processing framework designed to process massive datasets in a parallel fashion using many computing nodes. However, the use of MapReduce has declined in recent years, as this requires programming to perform even simple tasks. MapReduce has three significant steps:

- Map: Each worker invokes the map function and stores the result in temporary storage.
- Shuffle: Data is re-distributed across workers, so only data with the same key is stored on the same node.
- Reduce: Each group of output data is processed in parallel by worker nodes.

1) Map reduce architecture

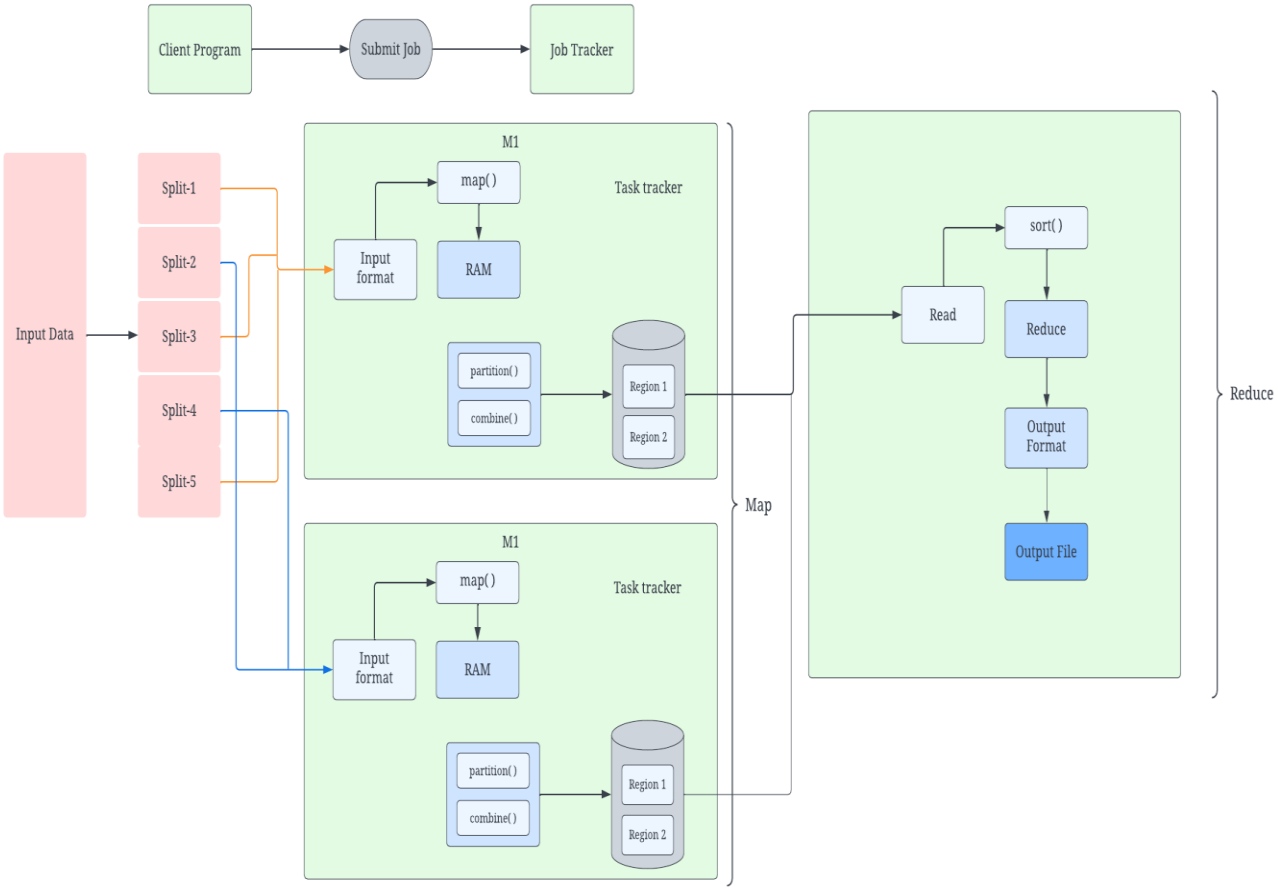

Fig. 5: Map Reduce [17]

In MapReduce, a job of a specific size will be submitted by the client to the MapReduce Master. This work will now be divided into additional equivalent job parts by the MapReduce master [17]. The Map and Reduce task will then have access to these job parts. This Map and Reduce task will include the procedure following the specifications of the specific business's use case. The developer writes the logic to meet the standards set by the sector.

The map task is then supplied with the input data we are utilizing, and the map will provide an intermediate key-value pair as its output. The Reducer receives the Map output (key-value pairs) and stores the result on the HDFS. It is possible to create n different Map and Reduce tasks to process the data as needed. The Map and Reduce algorithm have been carefully designed to have the least amount of time or space complexity.

B. Apache Hadoop

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

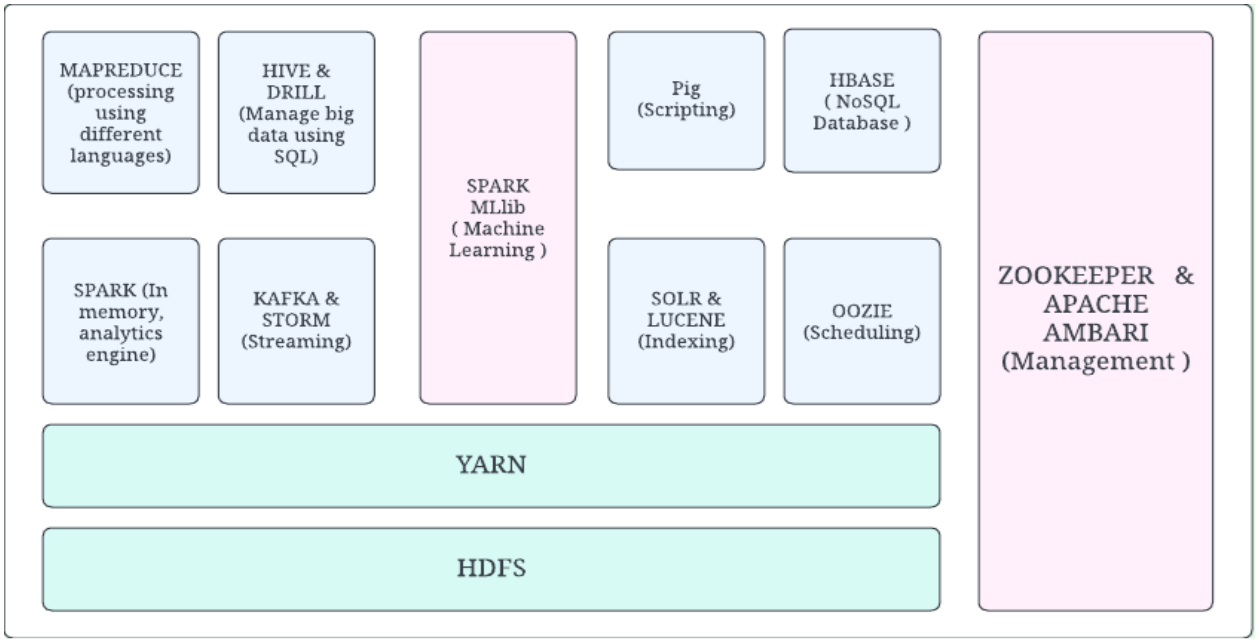

Fig. 6: Apache Hadoop [18]

Apache Hadoop is an open-source distributed data storage and data processing platform. Hadoop solves many problems in the big data domain. It accepts a variety of data from different sources, and it can be structured or unstructured. Additionally, Hadoop employs a cluster of commodity hardware to store data, making it economical because adding nodes is less expensive [18]. With Hadoop, a massive amount of data can be processed quickly due to its distributed architecture. Hadoop follows master-slave architecture, which makes it resilient to fault as the master node keeps track of all the child nodes via a messaging protocol called heartbeat, and if any child node goes down, it spins up a new node and replaces the failed node. Furthermore, Hadoop has a single NameNode and a single NameNode, so in the event of NameNode failure, standby NameNode takes its place, thereby making it fault tolerant. Finally, when a task is submitted to Hadoop, it is split into a number of sub-tasks and distributed to data nodes; this process breaks massive data into small chunks, leading to low network traffic.

C. Apache Spark

Large data sets can be processed quickly with Apache Spark, a data processing framework. Furthermore, it can divide data processing jobs across various computers on its own or in cooperation with other distributed computing technologies. Big data and machine learning, which need a lot of computational power to analyze data. Spark not only abstracts away from developers a large portion of the tedious work involved in distributed computing and big data processing, but it also lessens some of the programming burdens related to these activities [19].

Spark was created to overcome the limitations of MapReduce by doing processing in memory, minimizing the number of steps in a job, and reusing data across several concurrent operations. With Spark, the process of reading data into memory, performing operations, and writing out the results just requires one step, leading to significantly faster execution. Spark additionally reuses data by utilizing an in-memory cache to dramatically accelerate machine learning algorithms that often run a function on the same dataset .

Data can be reused by creating Data Frames, an abstraction over Resilient Distributed Datasets (RDD), which are collections of items cached in memory and used in numerous Spark operations. Due to the huge reduction in latency, Spark is now several times faster than MapReduce, especially when executing machine learning and interactive analytics.

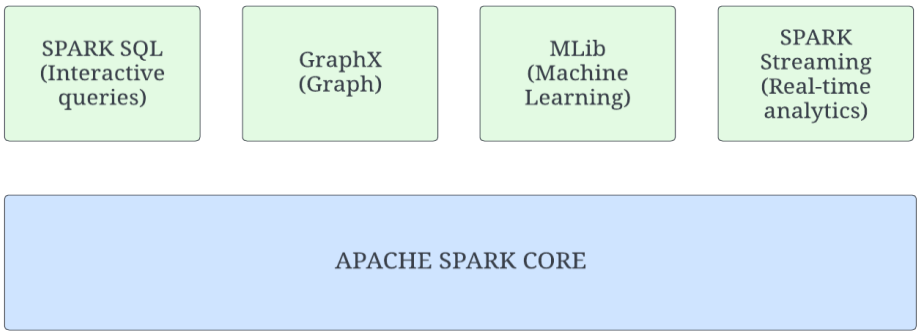

Fig. 7: Apache Spark [19]

1) Spark SQL

Spark SQL is a Spark module for processing structured data. On top of Spark Core, the Spark SQL component adds a new data abstraction called SchemaRDD, which supports structured and semi-structured data. It enables Hadoop Hive queries to be processed 100 times faster on current implementations and data. Additionally, it also provides strong compatibility with the rest of the Spark ecosystem.

2) Spark GraphX

A distributed graph processing platform called GraphX is built on Spark. GraphX provides ETL, exploratory analysis, and iterative graph computation to allow users to interactively construct and modify a graph data structure at scale. It comes with a variety of distributed Graph algorithms and a very versatile API.

3) Spark MLlib

Library of algorithms called MLlib is part of Spark and allows large-scale machine learning on data. On any Hadoop data source, data scientists can train machine learning models using R or Python, save the models using MLib, and then import the models into a Java- or Scala-based workflow. Machine learning can be carried out swiftly thanks to Spark, which was created for quick, interactive computing that operates in memory.

4) Spark Streaming

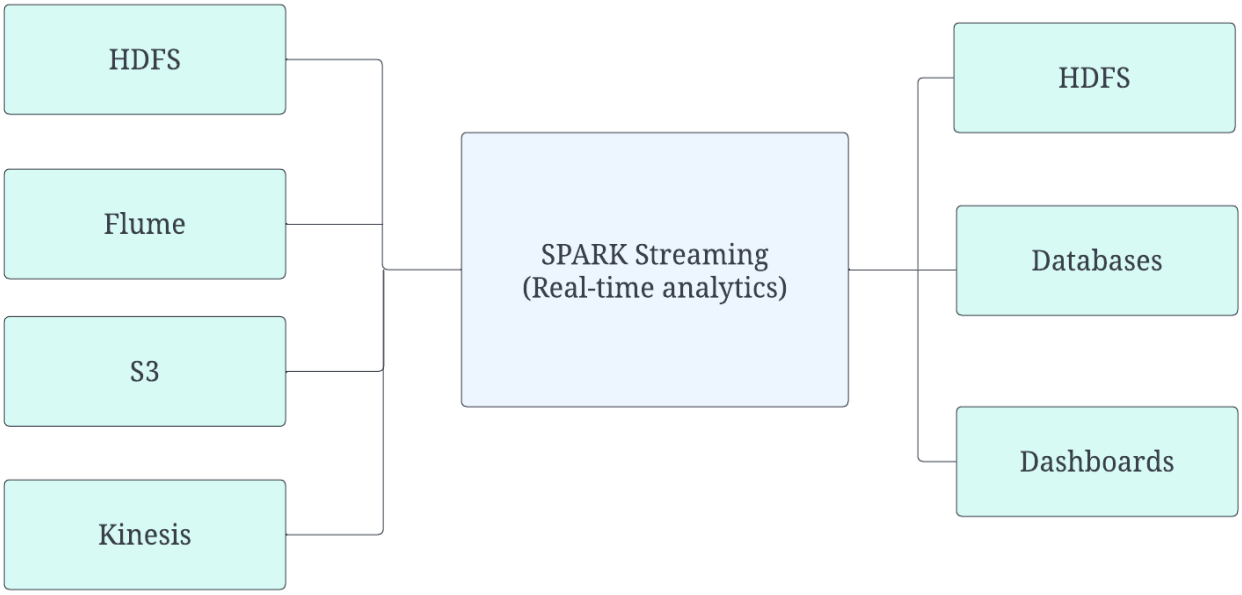

Fig. 8: Spark Streaming

One of the most crucial components of the Big Data ecosystem is Spark Streaming. The Apache Spark Foundation has created a software framework to manage Big Data. In essence, it continuously ingests data from sources, processes it with functions and algorithms, and then pushes the data out to be stored in databases and other places.

Spark Streaming, an addition to the Spark API, enables data engineers and data scientists to examine real-time data from numerous sources, including Kafka, Flume, and Amazon Kinesis. This transformed data can be distributed to databases, file systems, and real-time dashboards.

VIII. NOSQL DATABASES

NoSQL is a non-relational DBMS that is simple to scale, does not require a set schema, and avoids joins. Distributed data repositories with high storage requirements typically employ a NoSQL database.

NoSQL is used in big data and real-time online applications. For example, companies like Meta, Amazon, and Google everyday collect gigabytes of user data.

Depending on the data model, there are different types of NoSQL databases:

- Document databases: Document databases hold data in files that resemble JSON (JavaScript Object Notation) objects. Each document has pairs of fields and values. The values typically come in a variety of forms, including words, numbers, booleans, arrays, and objects.
- Key-value database: A key-value database is a simpler type of database that has keys and values for each object.
- Wide-column database: Wide-column storage uses tables, rows, and dynamic columns to organize data.
- Graph database: In graph databases, data is stored in nodes and edges. Edges track the connections between nodes, whereas nodes usually have data about people, locations, and things.

Some popular NoSQL databases:

\Table. 1: Popular NoSQL databases and their characteristics [20]

DB/ Properties	MongoDB	Cassandra	Couch DB	Hbase	Redis
Language	C++	Java	Erlang	Java	C, C++
Data Model	BSON	Big Table	JSON	Big Table and Dynamo	Data Structure
Fault Tolerance	Replication	Partitioning and replication	Replication	Partitioning and replication	Replication
Data Storage	Memory, file framework	Dynamo for storing data	Memory, File framework	HDFS	File system
Community	AGPL	Facebook	Apache	Apache	BSD
MapReduce	YES	YES	YES	YES	NO
Query Language		Java API, Thrift API	XML, Thrift API	API calls	JavaScript
Replication Modes	Master-Slave Replication	Master-Slave replication	Multi-Master Replication	Master-Slave Replication	Master Slave Replication
Protocol	TCP/IP	Thrift	HTTP/REST	Thrift, API, tradition	Binary, Similar to telnet

IX. APACHE FLINK

Flink is a scalable data analytics platform and distributed processing engine. Flink has been developed to operate in all typical cluster environments and to carry out calculations at any scale and in-memory speed.

Flink can be used in streaming applications to analyze data streams at scale and provide in-the-moment analytical insights on the processed data. Flink is made to operate in all typical cluster environments and handle calculations at any scale and in-memory performance. For distributed computations over data streams, Flink provides data distribution, communication, and fault tolerance [21].

The creation of Flink streaming applications with a data pipeline made up of one or more data sources, data transformations, and data sinks is possible with the use of APIs. In order to take use of the scalability and state management capabilities of Flink, we can design the architecture of your application with parallelism and windowing methods.

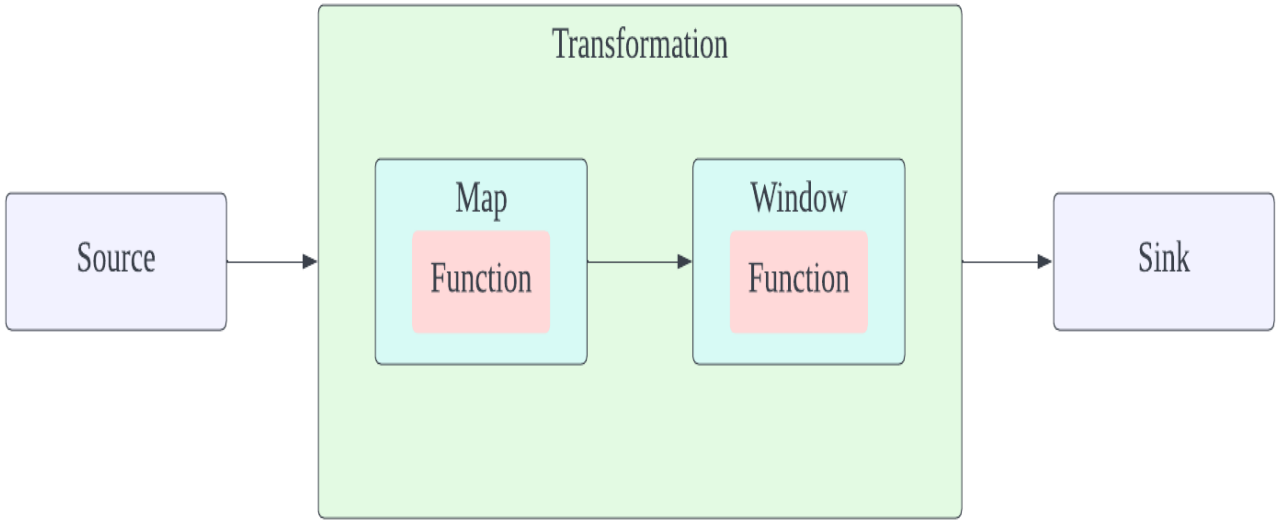

Fig. 9: Apache Flink data flow [21]

When creating Flink streaming applications, the DataStream API serves as the main API. Datastream and transformation are the main components of a streaming application. As shown in the figure 9, a Flink program transforms the incoming data streams from a source into one or more output streams sent to the sink.

The essential logic of a Flink application is built in a pipeline using the structure of the above data flow. A data flow allows for the definition of one or more operations that can be carried out concurrently and independently.

In addition, windowing functions allow for the application of various computations to various streams within the specified time window, further maintaining the processing of events.

X. BIG DATA ANALYTICS ARCHITECTURE USING AWS SERVICES

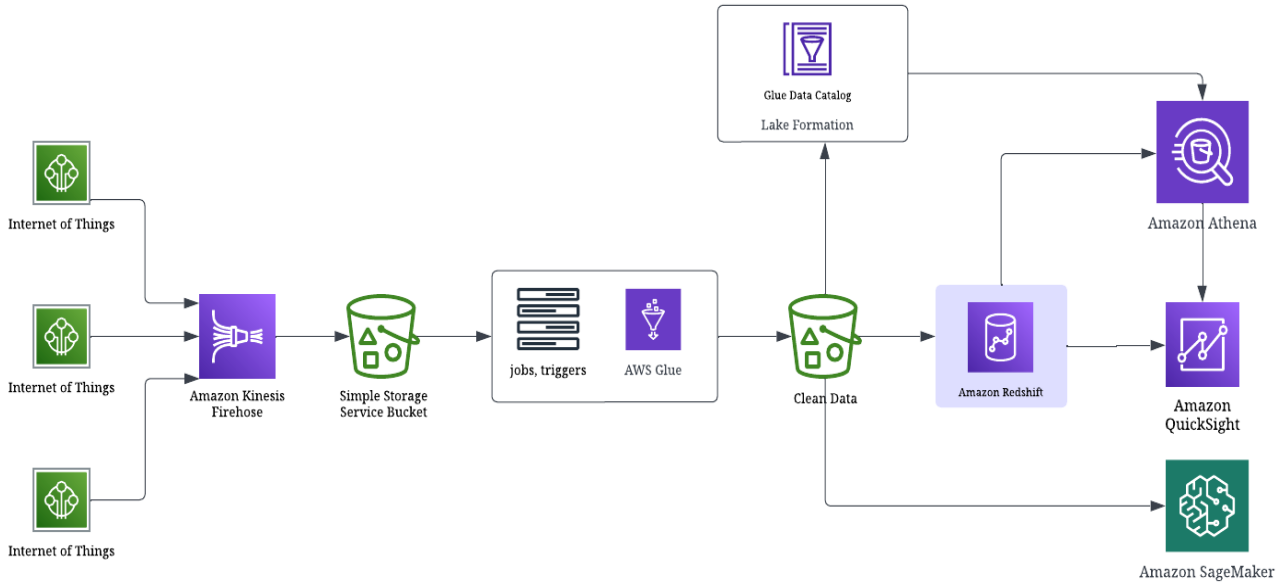

Fig. 10 Data pipeline using AWS services

In figure 10, data pipeline uses data produced by IoT devices, this data is ingested in real time using Kinesis Firehose and the data is dumped into an Amazon S3 bucket. At this stage data is extracted from S3 using glue and some transformations and data quality check is done and stored in the other S3 bucket. Now we have cleaned data which can be further utilized to derive business value out of it.

Furthermore, Large amounts of data stored in Amazon S3 can be easily analyzed utilizing serverless query services like Amazon Athena. Athena helps analyze data, gather statistics, or compile insightful summaries regarding data. We try to maintain our solution scalable and economically feasible when using AWS services. Additionally, it reduces the administrative burden for teams. For stream ingestion and processing, use the Kinesis family of services. Amazon Kinesis Data Streams may receive the streaming data from hundreds to thousands of IoT sources (including clothes and equipment) (KDS). KDS and Amazon Kinesis Data Firehose provides a way for buffering streaming data before it reaches Amazon Simple Storage Service (S3). SQL, Apache Flink, or Beam can be used to handle and analyze Kinesis stream data using Amazon Kinesis Data Analytics.

With the help of these services, we can easily create and execute powerful SQL queries using Amazon Kinesis Streams as your source. In addition, you can run time series analytics, provide data for real-time dashboards, and generate real-time metrics in this manner. The streaming data may need to be improved or transformed before being sent to Amazon S3. Data transformation is possible with an AWS Lambda function and Amazon Kinesis Data Firehose. The altered and unprocessed data will then be transmitted to the destination (Amazon S3) through Kinesis Data Firehose.

AWS Glue can be used for more complicated transformations. For instance, you can begin preparing and aggregating the training dataset using Amazon SageMaker Data Wrangler as soon as the data is on Amazon S3. Using Amazon SageMaker, a fully-managed service, data scientists can quickly and easily design, train, and deploy machine learning models at any scale. Utilizing modules from Amazon SageMaker that can be used independently or together, develop, train, and deploy your machine learning models.

XI. CONCLUSION

In this paper, the viability of enabling big data analysis in the context of the smart grid is discussed. We first highlight the significance of big data analysis in smart grids by identifying specific critical data needs for the upcoming grid. Then, a succinct introduction of a few new technologies that have recently appeared is provided. These technologies can be employed to build big data applications for the smart grid. Massive amounts of data are gathered by smart grid technology to make the grid more intelligent. The nature, distribution, and time-sensitive nature of the collected data offer problems for utilities simultaneously. The paper focused on the processing, storing, and even visualizing capabilities of Big Data technologies and approaches that might be applied to meet the demands of smart grids. A general overview of the opportunities, guiding ideas, and difficulties associated with data management in smart grids. Additionally, to have an effective and scalable data management system, we offered the implementation methods, and tools for Big Data technologies for smart grids.

The study shows that Apache Spark is better suited for batch data processing. On the other hand, Apache Flink can be utilized for use cases that demand low latency, where a delay of the order of milliseconds might have a significant impact.

Furthermore, due to its ability to handle rows and rows of data in real-time, which is not achievable with Apache Spark's batch processing approach, Flink maintains its competitive advantage. Due to this, Flink is quicker than Spark.

DECLARATION

Funding	No funding.
Conflicts of Interest/ Competing Interests	Not conflicts of interest to the best of our knowledge.
Ethics Approval and Consent to Participate	No, the article does not require ethical approval and consent to participate with evidence.
Availability of Data and Material	Open Source software documentation.
Authors Contributions	Dr. Sridhar S (Abstract, Introduction and Pipeline Architecture), Aditya Arya (Review of various Big Data Technologies and Databases, Assisted in building data pipeline architecture).
Code Availability	Not applicable.

REFERENCES

1. Teradata.com. (2021, June 08). Teradata. [Online]. Available: https://www.teradata.com
2. V. Sultan and B. Hilton, "Electric grid reliability research," Energy Informatics, vol. 2, May 2019. [CrossRef]
3. De Dutta, S., Prasad, R. Security for Smart Grid in 5G and Beyond Networks. Wireless Pers Commun 106, 261–273, March 2019. [CrossRef]
4. L. Yang, H. Xue, and F. Li, "Privacy-preserving data sharing in Smart Grid systems.", January 2015. [CrossRef]
5. E. De Santis, L. Livi, A. Sadeghian, and A. Rizzi, "Modeling and recognition of smart grid faults by a combined approach of dissimilarity learning and one-class classification," Neurocomputing, vol. 170, pp. 368-383, December 2015. [CrossRef]
6. Zhang, Y., Huang, T. and Bompard, E.F. “Big data analytics in smart grids: a review”: Energy Inform, August 2018. [CrossRef]
7. D. Yao, M. Wen, X. Liang, Z. Fu, K. Zhang, and B. Yang, "Energy Theft Detection With Energy Privacy Preservation in the Smart Grid," IEEE Internet of Things Journal, vol. 6, pp. 7659-7669, 2019. [CrossRef]
8. D. Syed, A. Zainab, A. Ghrayeb, S. S. Refaat, H. Abu-Rub, and O. Bouhali, "Smart Grid Big Data Analytics: Survey of Technologies, Techniques, and Applications," IEEE Access, vol. 9, pp. 59564-59585, 2021. [CrossRef]
9. E. N. Yilmaz, H. Polat, S. Oyucu, A. Aksoz, and A. Saygin, "Data storage in smart grid systems.", April 2018. [CrossRef]
10. Y.-J. Kim, M. Thottan, V. Kolesnikov, and W. Lee, "A secure decentralized data-centric information infrastructure for smart grid," IEEE Communications Magazine, vol. 48, pp. 58-65, November 2010. [CrossRef]
11. H. Daki, A. El Hannani, A. Aqqal, A. Haidine, and A. Dahbi, "Big Data management in smart grid: concepts, requirements and implementation," Journal of Big Data, vol. 4, 2017. [CrossRef]
12. A. Mohamed, S. S. Refaat, and H. Abu-Rub, "A Review on Big Data Management and Decision-Making in Smart Grid," Power Electronics and Drives, vol. 4, pp. 1-13, November 2019. [CrossRef]
13. B. Fang, X. Yin, Y. Tan, C. Li, Y. Gao, Y. Cao, and J. Li, "The contributions of cloud technologies to smart grid," Renewable and Sustainable Energy Reviews, vol. 59, pp. 1326-1331, 2016/06/01/ 2016. [CrossRef]
14. F. Luo, Z. Y. Dong, J. Zhao, X. Zhang, W. Kong, and Y. Chen, "Enabling the big data analysis in the smart grid.", November 2017
15. G. Liu, W. Zhu, C. Saunders, F. Gao, and Y. Yu, "Real-time Complex Event Processing and Analytics for Smart Grid," in Complex Adaptive Systems, 2015. [CrossRef]
16. Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges," IEEE Transactions on Smart Grid, vol. 10, pp. 3125-3148, May 2019. [CrossRef]
17. J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Commun. ACM, vol. 51, pp. 107–113, 2008. [CrossRef]
18. A.S. Foundation. (2022, May 08). Hadoop. [Online]. Available: http://hadoop.apache.org
19. A.S. Foundation. (2022, Oct 08). Apache Spark. [Online]. Available: http://spark.apache.org.
20. M. S. Kumar and P. J, "Comparison of NoSQL Database and Traditional Database-An emphatic analysis," JOIV: International Journal on Informatics Visualization, vol. 2, p. 51, 2018. [CrossRef]
21. A.S. Foundation. (2022, Jan 02). Apache Flink. [Online]. Available: https://flink.apache.org

AUTHORS PROFILE

Aditya Arya, Undergraduate Student in Department of Electrical and Electronics Engineering at MS Ramaiah Institute of Technology, Bangalore, Karnataka, India. He is aspiring researcher and have worked extensively on big data systems and cloud computing using AWS services. His area of research interest is Machine learning, Distributed Systems and Big Data Analytics.

Dr. Sridhar S, Associate Professor in Department of Electrical and Electronics Engineering at MS Ramaiah Institute of technology, Bangalore, Karnataka, India. He has more than 14 years of experience in teaching Electrical engineering. He consistently strives to create a challenging and engaging learning in which students become life-long scholar and learner. Strongly invested in upskilling himself through constant research. His area of research interest is Smart Grid, Renewable Energy Technologies and fault analysis in electrical machines.