ISSN No:-2456-2165

Telco Big Data Analytics using Open-Source Data Pipeline: Layers, Implementation and Conclusion

Dr. Chandrasekar Shastry and Abirami Thangavel

Abstract:- The current global health situation (primarily due to Covid-19) has encouraged a change in customer behavior toward the use of telecommunications services, which has increased data traffic. Telecommunications operators now have a golden opportunity to use Big Data Analytics (BDA) solutions to develop new sources of revenue. We encountered a number of difficulties when trying to set up a BDA project, particularly when it came to selecting the technical solution from the vast array of available tools and the governance methods for managing the project and the data. Most study papers on the telecommunications sector have not covered BDA project implementation from beginning to end. This paper's focus is on BDA telecommunications, the value of open-source data pipelines, architecture, implementation methodologies, and architectural levels. The study's last section offers practical BDA use cases for applications that support cost reduction and revenue generation. It appears that this effort will make it easier to implement BDA initiatives and give telecoms operators a better understanding of the crucial factors that need to be prioritized. This research that will help achieve this objective.

Keywords:- *Big Data Analytics, Open Source, Data Pipeline architecture, Implementation of open source Data Pipeline for BDA, Apache Kafka, Big Data Architecture Layers, Apache Kafka, Spark, Telecommunication, Telecom BDA Use Cases, Kubernetes for Telco data analytics*

I. INTRODUCTION

The telecommunications sector collects a great deal of information from various sources that needs to be transformed into some actionable insights, such as KPIs but also network statistics, hardware and software infrastructures, subscriber behaviors, data traffic nature, causes of service failure, analyzing mobile models for service connectivity, customer churn, demographic-based customer usage, gender-based usage, mobile data network usage, peak timings, and so on [40]. It is critical to immediately get some useful insights from these big data sets in order to support telco companies in expanding their businesses and decreasing customer turnover.

With the use of smartphones and other linked mobiledevices growing. The volume of data flowing across telecom operator networks has increased as a result of the rising use of smartphones and other connected mobile devices.

They must quickly store, evaluate, and gather insightful information from the data at hand. Big data analytics are useful in this situation. Big data may help telecom firms become more profitable by enhancing customer experience, enhancing network utilization and services, and improvingsecurity.

The telecommunications sector has access to new prospects because to big data. It can enhance service quality and enable more efficient traffic routing. Tele businesses can also spot fraud and take prompt action on it by monitoring call data records in real-time. In the end, this gives them a market advantage and helps them discover untapped possibilities.

Highly dependable and real-time data analytics are now possible thanks to the growth of IoT, opensource technologies, and artificial intelligence. The data is processed both in batch mode and in real-time. User clickstream, geographic user data, call record details, mobile network usage data, network monitoring, customer and subscriber profiles, VOIP data, and hardware data are notable examples of telco data [21].

Three Vs—velocity, volume, and variety—can be used to characterize telecommunications data [37]. BDA is a procedure that aids in gaining insightful knowledge from the unprocessed data that is ingested from many sources or domains [39]. BDA makes extensive use of the data analytics used in the telecom sector as well as other technologies like open source, IoT, artificial intelligence, machine learning, and so forth.

Big data is now crucial for advancing the telecommunications sector. Telecommunications providers may significantly enhance their services and make their users happier with the correct data analytics strategy. Big data analytics may help businesses and organizations make more informed decisions, provide better customer service, and run more smoothly.

Several telecom operators started BDA programmes over the past ten years, however they were unable to produce the desired results. In fact, according to a McKinsey study of 80 telecom operators that invested in BDA platforms, less than 8% of them have seen profits above 10%, and nearly a third have seen profits of less than 0%[1]. According to a 2015.

Gartner prediction, 60% of BDA projects will fail [55], primarily because of poor management, a lack of a clear vision, and a lack of available talents. According to Jacques Bughin's research, the following factors might significantly and favorably affect the return on investment of BDA projects: First, the architecture decision, which

ISSN No:-2456-2165

will affect the performance and scalability. Second, the ownership of theinitiative must originate from the top of the business. The governance model comes last and must include all facets of project and data governance. There are still no regulations governing BDA projects and data, according to other studies (Otto, 2011), (Zahid et al., 2019), and there is no reference design for telecom initiatives.

The goal of this study is to provide telecom players with a framework based on best practices that will allow them to secure the three most important components—project and data governance, solution architecture, and the project's necessary competencies—for the success of the implementation of their BDA projects. We undertook a review of the literature on BDA implementation in the telecom sector in order to get to this realization.

The main intent of this paper is to identify to what extent the potential BDA for telco has influenced the academic research. The reason to focus on academic research is that the fast-growing BDA landscape has left much space for some quality research to identify the impact of big data analytics tools for telecom. Another issue is the BDA process in telecommunications is complex. Each of the several complex tasks that must be carried out in the form of a pipeline can be completed with ease utilizing both open source and proprietary software [67]. Data upload, data transformation, data cleaning, and finally statistical and other analytics and visual communication are the main steps in big data analytics for telco. The learning curve differs for each opensource data pipeline architecture.

We formulate the following research questions for this purpose:

- RQ1: How much study material has been devoted to the architecture stack and big data analytics platform for the telecommunications industry?
- RQ2: Which of these architectures offers the most advantages, and which of the potential problems raised in the literature is addressed?
- RQ3: How can these possible issues be resolved when creating a big data analytics architecture for the telco industries?
- 4) RQ4: What prospective open-source components, if any, have been discussed in the literature and what proportion of these components make up the BDA data pipeline?

In order to investigate the aforementioned research issues, we followed a thorough process of evaluating the literature. This assessment specifically addresses the project and data governance techniques, architectures, and skill needs of the BDA. Next, we outline and analyze the most widely used methodology and architectural designs already in use by a number of telecom operators, as well as the pertinent skills needed to make such projects successful. We give a list of BDA use cases that have been effectively implemented in the telecom sector in the final section of this paper with complete implementation details for churn analysis along with the results.

II. WHY OPEN-SOURCE DATA PIPELINE FOR BDA

Nowadays, practically every business tries to be data-driven in some way. Businesses use data to better understand their customers, streamline corporate processes, and eventually increase profitability across all major verticals, including healthcare, telecommunications, banking, insurance, retail, and education. Businesses encounter two main obstacles when leveraging data for analytics: data tracking and connection establishment between business intelligence and data. Data tracking is the process of following the necessary data from many sources in order to gain insights from it. For many eCommerce organizations, tracking consumer activity information like logins, signups, transactions, and even clicks like bookmarks from platforms like mobile apps and websites becomes a problem. Data transformation and BI tool compatibility can frequently prove to be a significant barrier when data is gathered to establish a connection between business intelligence and data.

Businesses frequently only have two options when selecting a data analytics stack: buy it or develop it. On the one hand, there are solutions that are proprietary—like Google Analytics, Amplitude, Mixpanel, etc.—where the suppliers are solely in charge of configuring and managing them to meet your demands. Your main focus can be project management rather than technology management thanks to the best-in-class features and services that come with the tools. Open-source software makes up a large portion of the tools used for data analytics work. A programme is referred to as "open-source" if its public codebase is free to use. The majority of open-source products are started by programmers, then developed and improved by voluntary participants, although others are produced and maintained by nonprofit organizations. Across all expertise levels, developers, analysts, and engineers employ a large number of well-known open-source products. All types of labor can benefit from open-source options, and data analytics is no exception.

Following are the advantages of open-source technologies over proprietary software:

A. Cost-effective

Beyond their free tier, proprietary analytics systems can cost hundreds of thousands of dollars. The return on investment for small to medium-sized firms frequently does not outweigh these expenses.

In contrast to their proprietary counterparts, open-source technologies are cost-effective even in their business editions. Open-source analytics solutions are therefore far more cost-effective because they have lower upfront costs, fair prices for support, maintenance, and training, and no license charges. Furthermore, they offer superior value for the money.

ISSN No:-2456-2165

B. Flexibility

There will always be limitations on how proprietary SaaS analytics packages can be used. This is particularly true for the free trial or lite versions of the tools. For instance, certain tools don't support complete SQL. This makes it challenging to mix and query both internal and external data. Additionally, you'll discover that warehouse dumps frequently offer no assistance. When they do, they'll probably be more expensive and still just have a few functions. For example, only Google BigQuery may be used to load data dumps from Google Analytics. These dumps are also delayed in time. That implies that the loading time may be very long. You have complete choice when using open-source software: how you utilize your tools, how you integrate them, and even how you use your data. You can make the necessary adjustments if your requirements change—which, let's face it, they surely will—without paying extra for customized solutions.

C. No Vendor Lock-in

In essence, vendor lock-in, often referred to as proprietary lock-in, is a situation where a client becomes totally reliant on the vendor for their goods and services. The expense of moving to a different provider would be too high for the customer. Some businesses invest a substantial sum of money on proprietary products and services that they rely on extensively. The organization using these tools is seriously undermining its ability to compete if they are not regularly updated and maintained. Open-source tools nearly never experience this. The norm is constant innovation and change. The community can continue the project and maintain it even if the person or group managing the tool departs. When using open-source software, you can be sure that your tools are always up-to-date.

D. Privacy and Security

Recently, privacy has come up frequently in conversations involving data. This is due in part to the implementation of data privacy legislation like the GDPR and CCPA. The issue has also remained a top priority due to high-profile data exposures. You have total control over your data when using an open-source stack analytics running in your cloud or on- premise environment. You are then able to choose which data should be used when and how. It enables you to control whether and how third parties can use your data.

E. Collaboration and Community Support

The amount of helpful community support is among the most important benefits of open-source software. Resources include discussion boards for asking questions like Stack Overflow, GitHub projects with open-source code discussions, and Slack groups for data engineers and data analysts. Using open-source software entails being a part of a sizable community that is exploring and learning with you and is eager to share what they have discovered. As a result of this collective advancement, tools are continuously modified and improved.

F. Customization

With open-source tools, it can be challenging to find the correct assistance, but you typically have the freedom to change the code to suit your needs. These adjustments can include building an internal library for the analysts you work with or adding specific dependencies to your products. Open- source technologies' flexibility makes it possible to handle many different data processing problems. Although most licenses don't have tight usage restrictions, some of them forbid using the code for commercial gain, so be careful to check the project's license on GitHub.

G. Business Needs

It's not simple to choose the best instrument for data analytics. Every group, undertaking, and individual requires a data analytics solution that is customized for them. As we previously noted, a data team must devote a lot of time to building the necessary open-source knowledge to grow it throughout a company. However, if you're starting a business or have the time to invest in these technologies, open-source is a fantastic alternative. Some businesses provide open- source software with a paid support option. One illustration is dbt, which offers both free and paid subscription options. Paid support solutions combine the strength of open-source libraries and communities with the assistance of support personnel, which is exactly what some firms need to execute data analytics at scale.

After analyzing and documenting its big data strategy, an organization can choose the appropriate big data platform depending on its needs and features. Providing the ability to connect surreptitiously obtained and publicly accessible Big Data with information created within a corporation, as well as to deconstruct the joint set for value extraction, is a typical objective of such stages [13]. The following characteristics, in particular, should be included in big data platforms: A platform needs to be versatile and expandable in relation to the requirements and be in a complete, enterprise-ready state. Low computing power data updates are required, and a platform should be fault tolerant and of high quality. The platform can be both corporate and open source to facilitate development.

The four Vs (Volume, Velocity, Variety, and Value) are crucial in the data collection process. Data management, the following task, is storing data in any storage system that can handle massive data or enormous amounts of data. Big data applications and services can access, process, and use the data because it is stored in a format that makes this possible. Data analysis is done in the last step to find patterns, correlations, and other websites that are helpful. Before choosing and putting into use a big data platform, any agency, business, or individual with future intentions to use one should conduct research and develop a big data strategy. Big data platforms can aid consumers in understanding their needs [12].

ISSN No:-2456-2165

III. ARCHITECTURE

The suggested architecture is displayed in Fig 1. Various data sources are used to get the big data, which is shown in the the operations are carried out concurrently, and Spark processes all the data coming from several Kafka sources concurrently before passing it to a Logstash instance. MongoDB, a highly scalable and adaptable NoSQL database for large data applications, sends all of its events and logs to Logstash for collection. Data gathering and processing are done quickly with MongoDB. The Logstash instance receives the collected data next. All of MongoDB's events and logs are gathered by Logstash, which then quickly transforms and processes the data.

An application for distributed message processing called Apache Kafka gathers real-time data from numerous sources and processes it in real-time. By incorporating machine learning algorithms into its processing system, this also makes AI and machine learning conceivable. The message is then delivered to Spark for real-time batch processing. Data extraction, processing, and loading into elastic search are all made easier by Spark. Real-time analysis of massive amounts of data is made possible through elastic search. Elastic search has been utilized for telco data analytics. The outcomes are then shown in a dashboard (like Grafana). Below is a depiction of the whole data pipeline architecture made up of open-source components (see Fig 1). Many alternative open- source components can be used for batch processing and streaming, but we chose Spark due to its great speed and versatility. Each of these applications is contained within a Kubernetes (or other container runtime, such as Docker, Mesos) pod and is capable of dynamic scaling based on the data requirements. There is a high level of fault tolerance, highly distributed, highly scalable, and most significantly, the suggested data pipeline design is 100% Kubernetes-based because the Kubernetes master supervises all Kubernetes pods. The main benefits of using a container runtime, such as Kubernetes, are that they make applications lighter and do not require a guest operating system, unlike virtual machines. Since virtualization is crucial for cloud-native apps, our suggested approach is quite efficient for modern telecom needs.

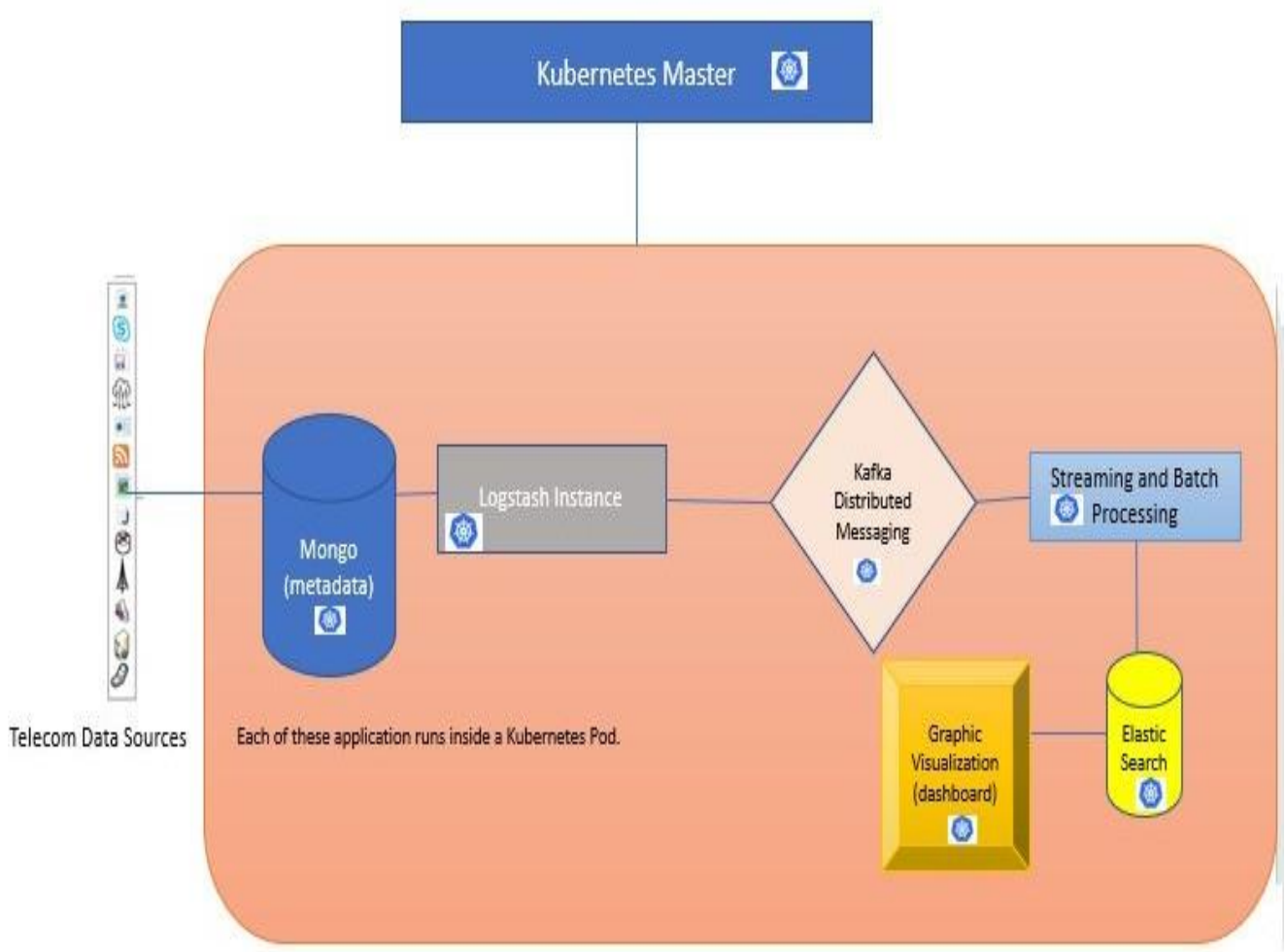

Fig. 1: Open-source data pipeline architecture

ISSN No:-2456-2165

IV. IMPLEMENTATION USING LAYERS

A. Connection Layer

The Connection layer enables our BDA pipeline to receive data from various telecom data sources. To put it another way, this layer implements an API of data connectors for a variety of regular SQL databases, No-SQL databases, IoT feeds, and other streaming or batch telecom data sources. The implementation of this layer is made easier by Python's support for connections to NoSQL and other databases, such as the pymongo API for connecting to a MongoDB instance and the redis-py API for connecting to a Redis database instance.

B. Integration Layer

The responsibility for combining telecom data from the Connection layer and adding it to an integrated data lake rests with the Integration layer. To store the data lake, we suggest deploying a master database (either on a single server or numerous servers). As a result of MongoDB's adaptable storage format and support for both batch and streaming data, we also recommend using it for this purpose. The actual data integration can be carried out by first saving each unique telecom source's data in the appropriate database (ideally a NoSQL one), and only then building a controller API to coordinate across these many stores.

For instance, newsfeeds from social networks may be continually saved in Neo4J and call detail records may be maintained in MongoDB, both of which are controlled by a controller that maintains meta-data about affiliations, data, and access-granting operations. In order to promote quicker recovery and storage with minimal administrative cost, Redis is our suggested metadata store. The tools from Talend and Pentaho cannot be used to incorporate data in this situation. Our suggestion is toward Python programming for each level in order to retain the effectiveness of the BDA process.

C. Batch Layer

Big telecom data from the master database is processed in batches (statically) by the batch layer. To complete the significant ETL processes for telecom large data in this layer, we advise employing a Hadoop cluster. Apache Spark can be used to complete activities that need to be completed quickly, while MapReduce can handle longer and more time-consuming operations. For example, MapReduce can compute the average call time for five years over 250 TB of data. To support the processing carried out in the streaming layer, the Batch layer offers a detailed drilled-down examination.

D. Streaming Layer

Real-time/dynamic) telecom feeds are processed by the streaming layer. This layer offers abstract real-time views and fundamental analytics. For consuming streams and processing them, we advise using Apache Kafka's SparkStreaming functionality. Flume, which is designed specifically for log analysis, is another Kafka rival, however Kafka's usecases far outnumber Flume's. Similar to SparkStreaming, Apache's Storm API is a rival, but SparkStreaming is thought to be more practical in terms of use cases. As BDA technologies advance, a concrete study should be conducted before choosing the tool. If any use case arises, high velocity dynamic data streams could be stored in MongoDB or Redis; otherwise, it could be ineffective [88].

E. Serving Layer

The results of the Batch and Streaming layers are combined at the Serving layer. It serves as a staging area where the outcomes of batch and stream processing are combined in accordance with the needs of end-users, such as C-level executives of the telecom business. We advise setting up the layer on a different server system, which we refer to as the analytical data lake. This layer sets processed data up for end-user dashboard display.

F. Interface Layer

The Interface layer joins the front-end layer with all of the back-end layers (Connection, Integration, Batch, Streaming, and Serving) (Dashboard layer). Here, well- known Python APIs for REST interface implementation, such as Dash and Flask, can be used. Also, because of its improved scalability, we advise choosing Node.JS web server technology.

G. Dashboard Layer

A number of dashboards are displayed on the dashboard layer for various telecom end users to view. Every dashboard connects to the Serving layer via common connections, such as BI connectors provided by different BI products (such Tableau, Oracle's NI, and QlikView), or Apache's Sqoop. To construct dashboards, Plotly, an open-source Python API, is helpful. Any of the well-known cloud service providers, such as Amazon's AWS, Microsoft's Azue, or Google's Cloud, could be used to deploy the entire pipeline (or just the front end), depending on the needs of the client. We advise using SSL technology to connect dashboards to the Serving layer through the Interface layer. Both layers can process parallel and exclusive data streams. ETL (data cleansing) exercises and statistical modelling should be included in both the static and dynamic layers. For example, large data predictive analytics can be referred to as BigML. Programming the data management modules in Python will help to construct the background processes for handling static and dynamic data during this process.

H. Workflow Management

Both layers are capable of processing parallel and separate data streams. ETL (data cleansing) exercises and statistical modelling should be included in both the static and dynamic layers. BigML, for example, can be used to refer to big data predictive analytics. Programming the data management modules in Python throughout this step will help to construct the background processes for handling static and dynamic data.

I. Session Management

This module maintains a stateless record of each activity's session throughout the BDA pipeline. We advise generating archives with regard to a windowing time because this is likely to generate a lot of big data rapidly on its own. Here, using Redis as the session database is what we advise. We can use Apache Cassandra or HBase if we need storage over a longer period of time.

ISSN No:-2456-2165

J. Cache Management

This module saves every activity's session in the BDA pipeline in a stateless fashion. We advise generating archives with respect to a windowing time because this will probably generate a lot of massive data itself quickly. Redis should be used as the session database in this situation, per our advice. If we need to store data for a longer period of time, we can use HBase or Apache Cassandra.

K. Log Management

Log management can be used to provide information for client logging, server preparation, and troubleshooting, as suggested by best practices. Administrators shouldn't neglect to include client clickstream data in sessions and logs. Flume can be used to acquire logged data and perform MapReduce jobs on it.

L. Queue Management

Queuing up the jobs (for example, in an Oozie instantiation) may be necessary due to the varied demands of analytical tasks at various events. For real-time data streaming, Kafka is the best data queuing solution. We advise RQ (Redis Queue) programming, which operates queues in Redis programmed in Python, for static data. If Kafka is not appropriate, real-time data can still be processed using RQ.

M. Resource Management

The various resources and activities in the BDA pipeline are coordinated by this module. Apache Zookeeper is the finest application for the job since it can identify master node and slave node failures and assist in fault recovery. Additionally, it offers interfaces for effectively and efficiently managing cluster resources.

V. CONCLUSION

According to our assessment of the present BDA technology stack, the aforementioned BDA pipeline is typical. Last but not least, for easier coordination with other Dockers and more effective processing, we advise constructing this pipeline in a Dockerized fashion, with each activity running in its own docker container. Additionally, BDA pipelining needs a development operations (DevOps) structure, with the typical Jenkins tool managing continuous integration and continuous deployment. For improved security and an identity and access management (IAM) solution, all created data and outcomes should be kept in private GitHub repositories.

To support our choice, it is crucial to talk about the advantages and disadvantages of both lambda and kappa architectures. We chose Lambda because all of the use cases for telecommunication analytics call for both batch-level and real-time studies (primarily due to the requirement of data cleaning and machine learning at the batch level). Analyses are computed instantly in a kappa architecture since there is no operational component for batch-level analyses. Let's talk about two significant LambdaTel use cases from the telecom sector as evidence: a) Customer Relationship Management (CRM): Call Detail Records (CDRs) of customers are sent into both the batch layer and the streaming layer concurrently (CDRs are streaming in nature).

The CDRs are pre-processed and cleaned in batch mode over the course of an hour, and then machine learning is used to extract significant customer categories. The serving layer is then supplied these parts. Real-time analytics at the streaming layer begin to provide fundamental data right away, like customer call throughput and average calling time per unit time, which are also fed to the serving layer. In order to give business decision makers the necessary CRM picture, real-time results are then displayed with reference to segments that are available from the batch layer. This use case can also be used for other machine learning applications, such as prediction of telecom KPIs (calling time, SMS per second, frequency of mobile data usage, revenue, classification, regression, and time series forecasting).

Other machine learning applications, such as prediction (classification, regression, time series forecasting), of telecom KPIs, can also be used this use case (calling time, SMS per second, mobile data usage frequency, revenue, sales, etc.) b) Customer Attrition: CDRs and historical attrition data are fed to the batch and streaming layers in real-time, whereas marketing data and competitors' data are only provided to the batch layer at set intervals. In order to anticipate customer churn, the batch layer runs ETL, cleaning all the data and sending the results to the serving layer. Basic attrition metrics for previously computed customer segments are presented in the streaming layer, and the results are integrated to offer attrition predictions for each segment. In various use cases involving marketing, cross-selling/up-selling, human resource management, and operational analyses, we can similarly demonstrate the requirement for batch-level analytics.

Additionally, lambda architecture maintains a good balance between speed and reliability along with a fault-tolerant and scalable design due to Hadoop (plus Spark) implementation at batch level, which provides an error-free data execution process due to the presence of batch layer. It's noteworthy to note that Kreps, who really introduced kappa, questioned the lambda architecture in a blog post from 2014 [91]. He claims that lambda increases the amount of coding necessary for batch-level ETL, which is mostly needed for machine learning. In various use cases involving marketing, cross-selling/up-selling, human resource management, and operational analyses, we can similarly demonstrate the requirement for batch-level analytics.

But over the past five years, coding procedures have become significantly more straightforward because to Python's success as a data science and big data language. We may need to repeat an execution or reprocess a batch of lambda code, but this may be accommodated by employing in-memory and/or columnar storage solutions, which have also advanced since 2014. A lambda architecture is still challenging to migrate or reorganize, but given the dearth of published lambda telecom architectures, we believe the time for this migration is still some way off. Instead, the priority right now should be to develop and use it. Without concentrating on ETL, the kappa use case enables the execution of real-time queries, either on real-time data or data. Lambda can be used to address these issues, allowing us to temporarily disable the batch layer to meet these needs.

REFERENCES

[1.] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *IEEE Network*, vol. 30, no. 2, pp. 54–61, 2016.

[2.] F. X. Diebold, "Big data dynamic factor models for macroeconomic measurement and forecasting," in *Advances in Economics and Econometrics: Theory and Applications, Eighth World Congress of the Econometric Society, "(edited by M. Dewatripont, LP Hansen and S. Turnovsky)*, 2003, pp. 115–122.

[3.] B. Violino, "How to avoid big data analytics failures," https://www. infoworld.com/article/3212945/big-data/how-to-avoid-big-data- analytics-failures.html, 2017.

[4.] P. Zikopoulos, C. Eaton *et al.*, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill Osborne Media, 2011.

[5.] Demirkan and B. Dal, "The data economy: Why do so many analytics projects fail?" [Online]. Available: http://analytics-magazine.org/the-data-economy-why-do-so-many- analytics-projects- fail/, 2014.

[6.] E, "The world according to linq," *Communications of the ACM*, vol. 10, no. 54, pp. 45–51, 2011.

[7.] F. J. Ohlhorst, *Big Data Analytics: Turning Big Data into Big Money*, 1st ed., Amazon, Ed. John Wiley & Sons, 2012.

[8.] C. M. Ricardo and S. D. Urban, *Databases Illuminated*, 3rd ed., Amazon, Ed. Jones & Bartlett Learning, 2015.

[9.] G. C. Deka, *NoSQL: Database for Storage and Retrieval of Data in Cloud*, Amazon, Ed. Chapman and Hall/CRC, 2017.

[10.] M. D. D. Silva and H. L. Tavares, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*, 2nd ed., Amazon, Ed. O'Reilly Media, 2013.W.-K.

[11.] Datafloq, "Top reasons of hadoop - big data project failures," https:// datafloq.com/read/top-reasons-of-hadoop-big-data-project-failures/ 2185, 2017.

[12.] Kumari, S. Tanwar, S. Tyagi, N. Kumar, M. Maasberg, and K.- K. R. Choo, "Multimedia big data computing and internet of things applications: A taxonomy and process model," J. Network and Computer Applications, vol. 124, pp. 169–195, Dec. 2018.

[13.] Manyika, M. Chui, M. G. Institute, B. Brown, J. Bughin, R. Dobbs, Roxburgh, and A. Byers, *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. McKinsey, 2011. [Online].

[14.] Available: https://books.google.com.pk/books?id=APsUMQAACA AJH. Daki, A. El Hannani, A. Aqqal, A. Haidine, A. Dahbi, and H. Ouahmane, "Towards adopting big data technologies by mobile networks operators: A moroccan case study," in *Proc. 2nd IEEE Int. Conf. Cloud Computing Technologies and Applications*, 2016, pp. 154–161.

[15.] T. White, *Hadoop: The Definitive Guide*, 3rd ed., Amazon, Ed. USA: Yahoo Press, 2012.

[16.] C. M. Murphy, "Writing an effective review article," *Journal of Medical Toxicology*, vol. 8, no. 2, pp. 89–90, Jun 2012. [Online]. Available: https://doi.org/10.1007/s13181-012-0234-2

[17.] Chih-Lin, Y. Liu, S. Han, S. Wang, and G. Liu, "On big data analytics for greener and softer ran," *IEEE Access*, vol. 3, pp. 3068–3075, 2015.

[18.] H. Park, H. Gebre-Amlak, B. Choi, S. Song, and D. Wolfinbarger, "Understanding university campus network reliability characteristics using a big data analytics tool," in *Proc. 11th Int. Conf. Design of Reliable Communication Networks*, March 2015, pp. 107–110.

[19.] S. Parise, "Big data: a revolution that will transform how we live, work, and think, by viktor mayer-schonberger and kenneth cukier," *J. Information Technology Case and Application Research*, vol. 18, no. 3, pp. 186–190, Sept. 2016. [Online]. Available: https://doi.org/10.1080/ 15228053.2016.1220197

[20.] D. Sipus, "Big data analytics for communication service providers," in *Proc. 39th IEEE Int. Conv. Information and Communication Technology, Electronics and Microelectronics*, May 2016.

[21.] Bughin, "Reaping the benefits of big data in telecom," *J. Big Data*, vol. 3, no. 1, 2016.

[22.] Rathore, A. Paul, A. Ahmad, M. Imran, and M. Guizani, "Highspeed network traffic analysis: Detecting VoIP calls in secure big data streaming," in *Proc. IEEE 41st Conf. Local Computer Networks*, Nov. 2016, pp. 595–598.

[23.] S. Han, C. -L. I, G. Li, S. Wang, and Q. Sun, "Big data enabled mobile network design for 5g and beyond," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 150–157, 2017. [Online]. Available: https://doi.org/10. 1109/mcom.2017.1600911

[24.] F. Hueske and V. Kalavri, *Stream Processing With Apache Flink: Fundamentals, Implementation, and Operation of Streaming Applications*, 1st ed., Amazon, Ed. USA: O'Reilly Media, 2018.

[25.] Garg, *Learning Apache Kafka, Second Edition*, 2nd ed., Amazon, Ed.USA: Packt Publishing, 2015.

[26.] J.-C. Tseng, H.-C. Tseng, C.-W. Liu, C.-C. Shih, K.-Y. Tseng, C.-Y. Chou, C.-H. Yu, and F.-S. Lu, "A successful application of big data storage techniques implemented to criminal investigation for telecom," in *Proc. 15th IEEE Conf. Asia-Pacific Network Operations and Management Symposium*, 2013, pp. 1–3.

[27.] T. Yigit, M. A. Cakar, and A. S. Yuksel, "The experience of nosql database in telecommunication enterprise," in *Proc. 7th IEEE Int. Conf. Application of Information and Communication Technologies*, 2013, pp. 1–4.

[28.] C. Şenbalcı, S. Altuntaş, Z. Bozkus, and T. Arsan, "Big data platform development with a domain specific language for telecom industries," in *Proc. High Capacity Optical Networks and Emerging/Enabling Technologies*, Dec. 2013, pp. 116–120.

[29.] Jonathan and K. Tor, "Subscriber Classification Within Telecom Networks Utilizing Big Data Technologies and Machine Learning," in *Proc. 1st Int. Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pp. 77–84, 2012.

[30.] C. Costa, G. Chatzimilioudis, D. Zeinalipour-Yazti, and M. F. Mokbel, "Efficient exploration of telco big data with compression and decaying," in *Proc. IEEE 33rd Int. Conf. Data Engineering*, 2017, pp. 1332–1343.

[31.] D. S. Yuri Diogenes, Tom Shinder, *Microsoft Azure Security Infrastructure (IT Best Practices - Microsoft Press)*, 1st ed., Amazon, Ed.USA: Microsoft Press, 2016.

[32.] B. R. Chang, H. F. Tsai, Z.-Y. Lin, and C. -M. Chen, "Access- controlled video/voice over ip in hadoop system with bpnn intelligent adaptation," in *Proc. IEEE Int. Conf. Information Security and Intelligence Control*, 2012, pp. 325–328.

[33.] SolidIT, DB-engines, ranking of key-value stores@ONLINE, 2017. [Online]. Available: https://db-engines.com/en/ranking/key-value+store, https://db-engines.com/en/ranking/document+store, https://db-engines. com/en/ranking/wide+column+store

[34.] M. Kiran, P. Murphy, I. Monga, J. Dugan, and S. S. Baveja, "Lambda architecture for cost-effective batch and speed big data processing," in *Proc. IEEE Int. Conf. Big Data*, 2015, pp. 2785–2792.

[35.] H. Zahid, T. Mahmood, and N. Ikram, "Enhancing dependability in big data analytics enterprise pipelines," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, G. Wang, J. Chen, and

[36.] L. T. Yang, Eds. Cham: Springer International Publishing, 2018, pp. 272–281.

[37.] W. Queiroz, M. A. Capretz, and M. Dantas, "An approach for SDN traffic monitoring based on big data techniques," *J. Network and Computer Applications*, vol. 131, pp. 28–39, Apr. 2019.

[38.] L. Zhou, A. Fu, S. Yu, M. Su, and B. Kuang, "Data integrity verification of the outsourced big data in the cloud environment: a survey," *J. Network and Computer Applications*, vol. 122, pp. 1–15, Nov. 2018.

[39.] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and X. Shen, "Internet of vehicles in big data era," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018.

[40.] J. Forgeat, "Data processing architectures — lambda and kappa," https:// www.ericsson.com/en/blog/2015/11/data-processing-architectures-- lambda-and-kappa, 2015.

[41.] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and X. Shen, "Internet of vehicles in big data era," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018.

[42.] J. ZAGELBAUM, "Kapp. architecture: a different way to process data," https://www.blue-granite.com/blog/a-different-way-to-process-data- kappa-architecture, Jan. 25, 2019.

[43.] S. Jain, M. Khandelwal, A. Katkar, and J. Nygate, "Applying big data technologies to manage QoS in an sdn," in *Proc. 12th IEEE Int. Conf. Network and Service Management*, 2016, pp. 302–306.

[44.] J. Forgeat, "Data processing architectures — lambda and kappa," https:// www.ericsson.com/en/blog/2015/11/data-processing-architectures-- lambda-and-kappa, 2015.

[45.] C. E. Perkins and P. R. Calhoun, "Authentication, authorization, and accounting (AAA) registration keys for mobile IPV4," *RFC*, vol. 3957, pp. 1–27, 2005.

[46.] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 190–199, 2015.

[47.] H. Zahid, T. Mahmood, A. Morshed and T. Sellis, "Big data analytics in telecommunications: literature review and architecture recommendations," in IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 1, pp. 18-38, January 2020, doi: 10.1109/JAS.2019.1911795.

[48.] E. J. Khatib, R. Barco, P. Muñoz, I. De La Bandera, and I. Serrano, "Self- healing in mobile networks with big data," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 114–120, 2016.

[49.] H. Isah and F. Zulkernine, "A Scalable and Robust Framework for Data Stream Ingestion," *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, 2018, pp. 2900-2905, doi: 10.1109/BigData.2018.8622360.

[50.] Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower son with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, 2014.

[51.] R. I. Jony, A. Habib, N. Mohammed, and R. I. Rony, "Big data use case domains for telecom operators," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom*, Dec. 2015, pp. 850–855.

[52.] J. Liu, F. Liu, and N. Ansari, "Monitoring and analyzing big traffic data of a large-scale cellular network with hadoop," *IEEE Network*, vol. 28, no. 4, pp. 32–39, Jul. 2014. [Online]. Available: https://doi.org/10.1109/ mnet.2014.6863129.

[53.] Saad, A. R. Amran, I. W. Phillips, and A. M. Salagean, "Big data analysis on secure VoIP services," in Proc. 11th Int. Conf. Ubiquitous Information Management and Communication. ACM, pp. 5, 2017.

[54.] J. Kreps. "Kafka : a Distributed Messaging System for Log Processing." In *Proc. Kreps2011KafkaA*, 2011.

[55.] R. Van Den Dam, "Big data a sure thing for telecommunications: telecom's future in big data," in *Proc. IEEE Int. Conf. CyberEnabled Distributed Computing and Knowledge Discovery*, 2013, pp. 148–154.

[56.] K. Wang, J. Mi, C. Xu, Q. Zhu, L. Shu, and D.-J. Deng, "Realtime load reduction in multimedia big data for mobile internet," *ACM Trans. Multimedia Computing, Communications, and Applications*, no. 5s, pp. 1–20, Oct. 2016. [Online]. Available: https://doi.org/10.1145/ 2990473

[57.] N.R Al-Molhem, Y. Rahal, and M. Dakkak. "Social network analysis in Telecom data", *Big Data 6*, pp. 99, 2019. [Online]. Availablet: https://doi.org/10.1186/s40537-019-0264-6

[58.] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimization for mobile networks toward 5G," *IEEE Network*, vol. 30, no. 1, pp. 44–51, 2016.

[59.] Y. Ouyang, L. Shi, A. Huet, M. M. Hu, and X. Dai, "Predicting 4g adoption with apache spark: A field experiment," in *Proc.16th Int. Symp. Communications and Information Technologies*, 2016, pp. 235- 240.

[60.] R. Siddavaatam, I. Woungang, G. Carvalho, and A.

Anpalagan, "Efficient ubiquitous big data storage strategy for mobile cloud computing over hetnet," in *Proc. IEEE Global Communications Conf.*, Dec. 2016, pp. 1–6.

[61.] R. Siddavaatam, I. Woungang, G. Carvalho, and A. Anpalagan, "An efficient method for mobile big data transfer over hetnet in emerging 5G systems," in *Proc. 21st IEEE Int. Workshop on Computer Aided Modelling and Design of Communication Links and Networks*, 2016, pp. 59–64.

[62.] J. van der Lande, "The future of big data analytics in the telecoms industry," *White Paper*, 2014.

[63.] Ö. F. Çelebi, E. Zeydan, O. F. Kurt, O. Dedeoglu, Ö. Ileri, B. AykutSungur, A. Akan, and S. Ergüt, "On use of big data for enhancing network coverage analysis," *ICT*, pp. 1–5,2013.

[64.] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *IEEE Network*, vol. 30, no. 2, pp. 54–61, 2016.

[65.] R. K. Lomotey and R. Deters, "Management of mobile data in a crop field," in *Proc. IEEE Int. Conf. Mobile Services*, 2014, pp. 100–107.

[66.] C.-M. Chen, "Use cases and challenges in telecom big data analytics," APSIPA Trans. Signal and Information Processing, vol. 5, pp. 12, 2016.

[67.] R. Siddavaatam, I. Woungang, G. Carvalho, and A. Anpalagan, "Efficient ubiquitous big data storage strategy for mobile cloud computing over hetnet," in *Proc. IEEE Global Communications Conf.*, Dec. 2016, pp. 1–6.

[68.] Drosou, I. Kalamaras, S. Papadopoulos, and D. Tzovaras, "An enhanced graph analytics platform (gap) providing insight in big network data," *J. Innovation in Digital Ecosystems*, vol. 3, no. 2, pp. 83–97, 2016.

[69.] C.-M. Chen, "Use cases and challenges in telecom big data analytics," APSIPA Trans. Signal and Information Processing, vol. 5, pp. 12, 2016.

[70.] X. Lu, F. Su, H. Liu, W. Chen, and X. Cheng, "A unified OLAP/OLTP big data processing framework in telecom industry," in *Proc. 16th IEEE Int. Symp. Communications and Information Technologies*, Sept. 2016, pp. 290–295.

[71.] S. B. Elagib, A.-H. A. Hashim, and R. Olanrewaju, "CDR analysis using big data technology," in *Proc. IEEE Int. Conf. Computing, Control, Networking, Electronics and Embedded Systems Engineering*, 2015, pp. 467–471.

[72.] Z. Sheng, S. Pfersich, A. Eldridge, J. Zhou, D. Tian, and V. C. M. Leung, "Wireless acoustic sensor networks and edge computing for rapid acoustic monitoring," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 64–74, 2019.

[73.] M. S. Parwez, D. Rawat, and M. Garuba, "Big data analytics for user activity analysis and user anomaly detection in mobile wireless network," *IEEE Trans. Industrial Informatics*, 2017.

[74.] M. S. Parwez, D. Rawat, and M. Garuba, "Big data analytics for user activity analysis and user anomaly detection in mobile wireless network," *IEEE Trans. Industrial Informatics*, 2017.