International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Diving Into Data Lakes

Kartikay Sharma[1], Shreya Patoa[2]

[1]JRE Group of Institutions, School of Computer Science and Engineering, Knowledge Park IV, Greater Noida, Uttar Pradesh 20131, India

[2] JRE Group of Institutions, School of Computer Science and Engineering, Knowledge Park IV, Greater Noida, Uttar Pradesh 20131, India

Abstract: ***The idea of a Data Lake is emerging as a popular approach to manage and build up the coming era of frameworks to ace new big data challenges, and there are many concerns and inquiries for large enterprises to implement data lakes. The paper discusses the idea of data lakes and offers the thoughts of the authors on the subject.***

Keywords: Data Lakes, Schema on read, Data swamps, ETL.

1. Introduction

Organizations have contributed a lot of time and resources into building Data Warehouses in the past. This exertion was done to recognize all the data required for analysis and reporting, characterizing the data model and database structure, and developing programs. The sequence of steps known as ETL (: Extract source data, Transform it, and Load it into the data warehouse) is regularly followed in the process. Altering the existing data warehouse requires a substantial amount of additional investment to redesign the programs that extract, transform, and load data – the most complex and expensive errand is the development of the ETL layer.

One of the major difficulties confronted by organizations today is that they require data to reveal not just what occurred in the past, but also what is likely to occur in the future. They look for prescient and significant insights, recovered from a variety of data through both batch and real-time processing handling to illuminate their strategies.

Conventional data warehouses fail to become the perfect solution to this challenge as they are hard to change, expensive to work with, and they cannot be scaled cost-efficiently to process the regularly growing volume of information. Data Lakes can help to fill these lacunae.

2. Understanding Data Lakes

James Dixon – the Chief Technology Officer at Pentaho – has, for the most part been credited with authoring the expression “Data Lake”. He draws comparison of a data mart (a subset of a data warehouse) with a water bottle - washed down, bundled and organized for simple utilization - while a data lake is more similar to a water body in its common state. Data originates from streams (the source systems) and proceeds towards the lake. Clients approach the lake to look at, take tests or make a plunge.

Data lakes can be accurately understood as a repository containing humongous volumes of raw data, in original formats, while allowing varying users to fetch and study that data as required. A Data Lake is a central container where data is retained, irrespective of its origin and organization. It is likely to contain structured data associating with relational databases (tables – rows and columns), semi-structured data (logs, CSV, XML), unstructured data (e-mails, documents, PDFs) and binary data (images, audios, videos). An assortment of storing and handling tools – generic with tools of the extended Hadoop ecosystem – are eligible for utilization for quick retrieval from a Data Lake.

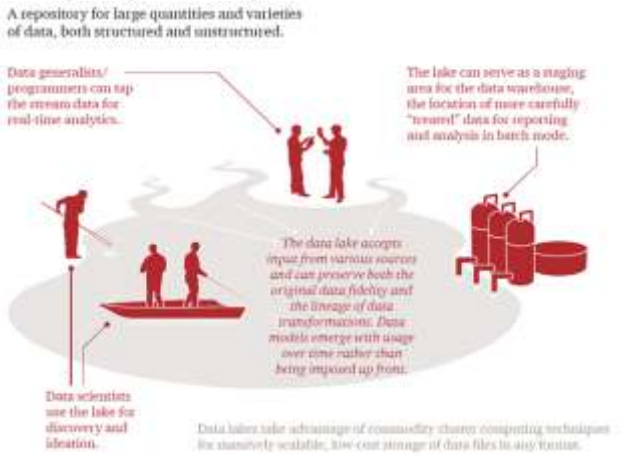

Figure 1: Data Lake [1]

Data lakes traditionally (until suggested otherwise), are constructed with the help of Hadoop. Business organizations can erect them using Infrastructure-as-a-Service (IaaS) clouds including Amazon Web Services (AWS) and Microsoft Azure. Amazon's Elastic Compute Cloud (EC2) performs maintenance of data lakes, whereas Microsoft devotes Azure Data Lake platform to retain as well as process real-time data.

2.1 Architecture of a Data Lake

Entire content shall be consumed by the data lake whole or staging repository, and then searched (with the help of a suitable search engine such as Cloudera Search or Elasticsearch). When required, content is to be analyzed, for the result to be now produced again to users through search to a variety of UIs across several platforms.

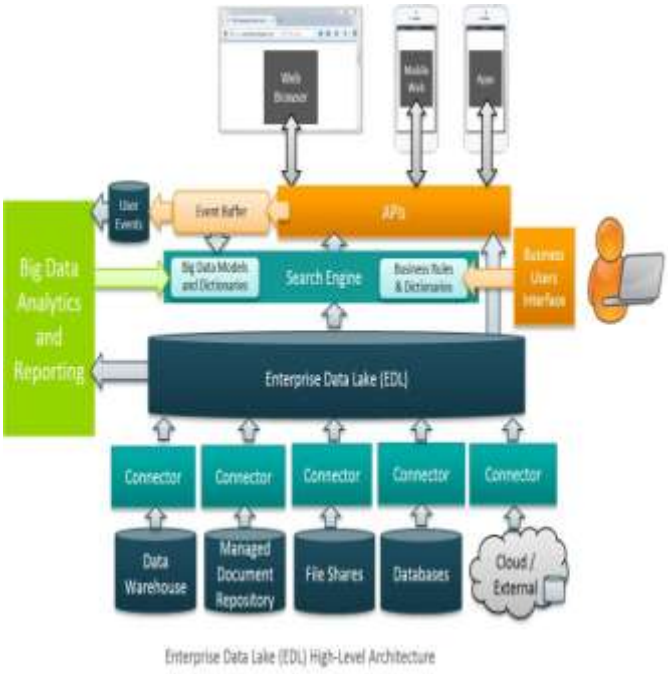

Figure 2: Architecture of Data Lake [2]

2.2 Five Principles of Data Lakes

EMC explains a Data Lake with five foundation pillars viz. **I**ngest, **S**tore, **A**nalyze, **S**urface and **A**ct, whose acronym ISASA is easy to memorize. These are fundamentals to strategize a data lake and hence support these functionalities. A breakdown of these principles is as follows:

1. Ingest = Collect data from multiple sources
This is accumulation of target data. Prerequisite of the systems to correctly and regularly ingest that data using APIs or batch processes enhances the data lake's proficiency.

2. Store = Scalable storage, with multi-protocol access
Centralizing data and breaking down silos are necessary for data lakes. Functionality is further enhanced if mountable storage and multiprotocol access can be delivered to the data. HDFS and CIFS are two examples for the same.

3. Analyze = Finding relationships with types of data
Establishing a relation between data points is of prime importance. This is should be done skillfully to ensure no data points are left behind without a relation.

4. Surface = Methods in visualizing the results of analysis
Requirement of a simple technique of showcasing all of the analysis and this data needs to be easy to understand. Enhanced visibility of results leads to ease in taking an action.

5. Act = Planned business-driven actions
In simpler words, it's the 4-Ms – Make Me More Money. A plan is devised to pick data analysis results and fit it into a functioning business model.

2.3 Building a Data Lake

The majority of data lakes are a product of incremental growth and experimentation. A few people know the knowledge of constructing a data lake. The correct method to create a data lake is to consider the white paper method: follow the data. In other words, follow your data.

The starting point distinguishes the path towards the data lake. Is the business completely aimed on big data? Or is big data merely a new concept? Does the company come from a data-driven, analytics culture background? Or is it muscle building regarding exploiting data?

To start a data lake, many companies employ the following strategies.

Stage 1: Handling data at scale
The first step is attainment of plumbing in place and realizing to acquire and change data at scale. Here, the analytics could be simple, but learning is done for getting Hadoop to work in the required manner.

Stage 2: Building transformation and analytics muscle
The second step is improving the facility to convert and analyze data. Here, companies and associated tools of desired skillset and further begin to collect more data and constructing applications. Capabilities from the enterprise data warehouse and the data lake are used collectively.

Stage 3: Broad operational impact
The third step is bringing data and analytics to maximum people. Here the data lake and the enterprise data warehouse work in collaboration, accordingly. One example of requirement for this blend is the reason that most big data companies that started with a data lake ultimately made an addition of an enterprise data warehouse to make the data operational. Likewise, companies with enterprise data warehouses don't abort them to favor Hadoop.

Stage 4: Enterprise capabilities
In this utmost level of the data lake, an addition of enterprise capabilities is made to the data lake. Selective companies have attained this level of maturity, but more will soon as the employment of big data breeds, requiring governance, compliance, security, and auditing.

2.4 Isn't Data Lake just Data Warehouse revisited?

In short – no, a data lake isn't just the data warehouse revisited. Instead, the only resemblances between them are that both are data storage repositories and their goal is to extract value from the data stored.

Now the question is whether data lakes are dissimilar from data warehouses. The table below focuses on a few of the major differences between a data warehouse and a data lake. This is, by no means, is an exhaustive list, but it does get us past the "been there, done that" mentality.

DATA WAREHOUSE	vs.	DATA LAKE
structured, processed	DATA	structured / semi-structured / unstructured, raw
schema-on-write	PROCESSING	schema-on-read
expensive for large data volumes	STORAGE	designed for low-cost storage
less agile, fixed configuration	AGILITY	highly agile, configure and reconfigure as needed
mature	SECURITY	maturing
business professionals	USERS	data scientists et. al.

Figure 3: Data Warehouse vs Data Lake [3]

Five denominators of difference between a data lake and a warehouse approach are as follows:

1. Data Lakes Retain All Data

While construction of a data warehouse, significant time is utilized for analyzing data sources, studying business processes and profiling data. The outcome is a supreme organized data model meant for reporting. This process majorly includes making choices about whether data is to be included in the warehouse. Usually, data may be rejected from the warehouse, if it is not capable to produce query results. This is done to streamline the data model, and also retain disk space, further for the data warehouse to be performant.

Comparatively, the data lake holds *all* data that can be utilized today or in future. Data is also retained at all times, for reference at any point of time for any analysis.

This is a feasible because hardware for a data lake is different from that utilized for a data warehouse. Commodity, off-the-shelf servers paired with cheap storage makes scaling a data lake to terabytes and petabytes inexpensive.

2. Data Lakes Support All Data Types

Data warehouses usually comprise of data obtained from transactional systems and consist of calculable metrics and the characteristics that describe them. Unconventional data origins like as web server logs, sensor data, social network activity, text and pictures are rejected commonly. Better use of such data types are discovered, however storing and utilization of it is not economical or feasible. The data lake method encourages these unconventional data types. In the data lake, we keep all data not in association of storage and structure used. Storage is in unprocessed form and transformed only when required for use. This methodology is "Schema on Read" vs. the "Schema on Write" used in the data warehouse. [See the graphic below.]

POS DATA	CRM	FINANCIAL DATA	LOYALTY CARD DATA	TROUBLE TICKETS
EMAIL	PDF FILES	SPREAD-SHEETS	WORD PROCESSING DOCUMENTS	RFID TAGS
GPS	WEB LOG DATA	PHOTOS	SATELLITE IMAGES	SOCIAL MEDIA DATA
BLOGS	FORUMS	CLICK-STREAM DATA	VIDEOS	XML DATA
MOBILE DATA	WEBSITE CONTENT	RSS FEEDS	AUDIO FILES	CALL CENTER TRANSCRIPTS

Figure 4: The data warehouse can only store the orange data, while the data lake can store all the orange and blue data.

3. Data Lakes Support All Users

80% or more of users are "operational", in many organizations. Demand of reports, crucial functioning metrics or slicing the same set of data in a spreadsheet daily, are some requirements. The data warehouse is commonly perfect for these consumers because it is thoroughly organized, easy to use and understand and it is fit for desired format of query reports generation.

The other 10%, desire further analysis on the data. Data warehouse is used as a foundation but traditional source systems are referenced to obtain data that excluded from the warehouse and often import data exterior to the organization. Conventional tools like the spreadsheet are used and they build latest reports that are further circulated amongst the organization. The data warehouse is one stop source for data but they often go afar its confines.

Lastly, a minor percent of consumers prefer skin-deep analysis. On the basis on research, new data sources may be generated. They pulverize various data types and spring up with exclusively new queries. Although they use the data warehouse but often dismiss it, as they are keen to indulge in abilities of moving afar from its boundaries. Such are the Data Scientists who use advanced analytic tools and abilities like statistical analysis and predictive modeling.

The data lake methodology upkeeps mentioned used well. The data scientists can visit the lake and work with the humongous and diverse data sets they need, while other consumers utilize structured views for data they have on hand.

4. Data Lakes Adapt Easily to Changes

Transformation time taken by data warehouses is a major drawback. Up front development and fabrication of correct structure is a time taking process. A suitable warehouse design is flexible but due to the complexity of the data loading process and the efforts to make analysis and reporting easygoing, these changes exhaust developer resources and is time taking.

Several business subjects can't invest greater time required by warehouse team who make the system flexible as per new demands for queries. The soaring need for instant results gave birth to the concept of self-service business intelligence. Whereas in data lakes, since storage of all data is in unprocessed form and easy to fetch on requirement, users are approved to go outside the warehouse structure to discover data in pure ways for query resolution.

In case of a productive outcome for an exploration process, which may be repeated, an exclusive schema could be used on it and automation & recycling is developed to serve the outcomes to a wider audience. If the outcome is not useful, it can be rejected and no modifications to the data structures have been made nor development resources consumed.

5. Data Lakes Provide Faster Insights

This last dissimilarity is the summary of other four. As data lakes comprises of all data and data types, because it allows consumers users to fetch data before transformation, cleansed and structured it provides results faster than data warehouse approach to the consumers.

Still, this early access costs capital. The work normally undergone by the data warehouse development team may not be done for many of the data sources necessary for study. Consumers hence survey and process data they consider appropriate but the first tier of business users mentioned previously might not adapt this process. Reports and KPI's are still required by them.

In the data lake, functioning report consumers utilize further structured views of the data in the data lake that mirror the data warehouse contents. These views are essentially kept as metadata that resides on the data in the lake other than physically rigid tables that require flexibility from a developer, is what sets it apart.

2.5 Advantages of a Data Lake

All industries and organisations can utilize and make profits from a data lake. *A data lake can abolish data silos* within an organization, bring all the data at a central location and obtain finer access to all disparate data sources within the business.

Access to 360-degree views of customers is widely demanded, and studying social media, but benefits healthcare organizations by optimizing treatments and allows manufacturers to produce insights from sensor data. Countable advantages of data lakes are as follows:

1. Low Cost, Extremely Scalable Storage

Storage in a data lake is economic and it can be scaled to humongous volumes.

2. Supporting Multiple Programming Languages and Frameworks

With primitive form of data in the data lake, developers can work with various programming languages such as Python or Java and use several frameworks such as Hive or Pig.

3. Data Agnostic and Immediate Access to All Data

Any data can be withheld in a data lake, on the spectrum of structured machine data to unstructured social media data at a single location. Furthermore, consumers can access all data at an instant, due to previously mentioned, which is necessarily role-based.

4. Centralized Data that does not have to Be Moved

Data is in a single location within a data lake. Silos can be rejected, allowing convenient access to pulverised data sources. Also, it eliminates the need to transfer data within warehouses.

5. More Insights Due to Raw Data

With the help of a data lake, organizations can retain the data in primitive format, ensuring data losses don't occur. In the future, with advancing functionalities associated to data, companies can always refer to previously stored data.

2.6 Challenges of a Data Lake – Avoiding Data Swamps

Data lakes have a few cons too. Before utilization on data lakes on larger scales some difficulties need resolution. Successful problem resolution will ensure that data lakes don't convert into swamps.

1. Meta Data Management

A tagged and catalogued data lake is dominantly best for an organization's use. Tagged data allows better queries and better study. For example, metadata is a fundamental constituent of a data lake. Metadata provides context, which of utmost importance for a data-driven world whose predecessors are varying data sources.

Twitter has command over it. Every tweet fetches 65 data elements that deliver background for each tweet. Metadata allows collection and pulverization of data to accomplish insights capable to transform business. However, consuming right data in the required time period is an eventful task. Adding metadata the instant data is input in the data lake is advisable yet an uncommon practice.

2. Data Governance

Data governance for an organization is challenging for dealing with data in general and big data more precisely. But for data lakes, it is of prime importance. But if ignored, when starting with a data lake, you can enter 'data limbo land' where varieties of issues associated with data quality, metadata management or security could spring up, leading the data lake to crash and fail.

The correct procedures in the organization ensure accurate data governance, and also ensuring that the right data is accumulated suitably and the right and correct algorithms are employed for data analysis.

3. Data Preparation

Proper handling of data is another dilemma. Progressing democratized entry to the lake is permissible and self-service becomes common, alternatives to refer to data quality and planning turn more critical.

Proper data preparation can be done by: 1) using appropriate data scientists who can analyze and fabricate sophisticated analytics models while confirming to data quality and data lineage or 2) Utilization of a system that formulates the raw data (semi-) automatically and hence allowing end-users to easily query the data and/or import in different analytical tools to achieve insights from the data.

The first method might not be feasible, because hiring, or training, data scientists isn't economic and not a favorable approach for the organization. The second method is rather feasible, since Big Data vendors might produce automated means by developing associated tools.

4. Data Security

Security is of concern due to a centralized data location. Organizations have faced data breaches in the past and it could hamper business against data lake breaches. Thus, the 'standard' security events should be used.

However, central data storage is a concern in terms of security. Thus, role-based entrance is critical when constructing a data lake. The data lake should ensure that although utilization of a central data location, entry is on terms of authority. To implement this, one can for example tag metadata with security data to ensure that role-based access is also implemented on the metadata level.

2.7 Technologies surrounding Data Lakes

The following picture provides different Hadoop technologies that fit into a data lake at a glance useful for deep analysis and is a jumping-off point. These are merely a common set of possibilities from the countless possibility of related technologies.

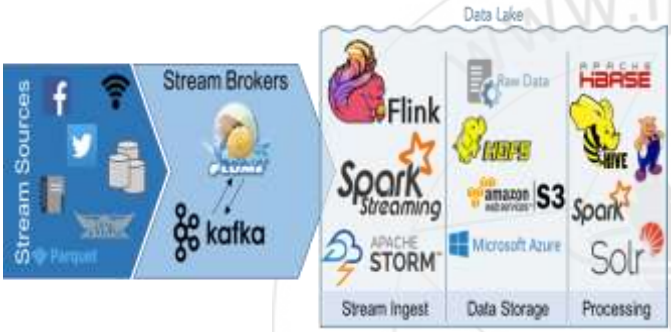

Figure 5: Technologies surrounding Data Lakes [4]

Data ingestion

For streaming in the data lake, the first pointer is that although several flexible technologies are present at hand and can be used in numerous contexts, a well-executed data lake provides confined rules to adhere to and methods around consumption. For example, Kafka and Flume both allow non-stop relations into Hive and HBase, and Spark can consume and transfer data without disk utilization by writing into it. This method is robust, but also puts the primitive, unchanged data, at a stake, which is a foundation of data lake design. Hence, data flow across the system is restricted. Data must be consumed, written at primitive handling zone for a grip, and hence mirrored to a different zone for transformation and development.

Flume and Kafka are giants of messaging systems used today. A layman viewpoint study of mention products is as follows:

Kafka is the latest amongst mentioned technologies, but has gaining power tremendously as a vigorous, scalable and fault-tolerant messaging system. Whereas Flume can be better described as a channel between two ends, Kafka is similar to a broadcast, allowing authorized consumers access to data "topics". Hence Kafka gains an edge over Flume in terms of scalability, and also by providing data redundancy and fault tolerance procedures. In case of failure of one Kafka, re-broadcast can be taken over by another Kafka. The area where Kafka lacks is commercial support. As of now, Cloudera incorporates Kafka, but MapR and Hortonworks don't. Furthermore, Kafka does not contain integrated connectors to other Hadoop yields. Some have been scripted, but generally, the same competence of "out-of-the-box" connectivity, as Flume can't be anticipated. *Flume being the only dominant streaming ingest choice*, it is deep-rooted in the Hadoop ecosystem backed up by all commercial Hadoop distributions. For large, enterprise-wide Hadoop deployments, it stands wholesome or even existing prerequisite features are a reason for choice. With emerging Hadoop technologies, Flume has defied its aging. Flume is a push-to-client system and operates between two locations instead of acting as a broadcast arrangement for consumers. Data will be gone in case of a Flume failure, without the reproduction of events, which is a con.

It may perhaps be observed without straying too far afield from our primary focus that Kafka and Flume do allow establishment of connectivity between each other, i.e. they are not certainly mutually exclusive. Flume comprises of both a sink and a source for Kafka, and there are various documented cases of establishing a link between the two, even in wide-ranging, fabrication systems. For small-scale or early-stage systems, unless there is a enthralling and ostensible need for both, it's advisable to select a single system on basis of current and potential needs.

Data processing

On having assembled a stream of data leading the Data Lake, options exist to arrange the data into a storable, consumable form. With Flume, writing directly into HDFS with built-in sinks is achievable. Kafka, however, does provide built-in connectors. Supplement of a stream-processing structure within the Hadoop ecosystem, can however profit both systems alike. Some frameworks are listed as follows.

Storm is a precise real-time processing structure, ingesting a stream as an exclusive "event," instead of small batch series. Meaning that Storm has very low expectancy and is eligible for consumption as a standalone single entity. Storm has been used in production occasions for lengthiest of the three mentioned solutions, but has commercial support at its disposal. Nevertheless, Storm does lack YARN support (it can be run on Mesos or as a Slider process on YARN), and single processing of data can't be guaranteed.

Spark is popular for it's in-memory treating competences and the Spark Streaming unit runs on similar basis. Spark isn't a pure "real-time" scheme. In its place, micro batch processing at pre-defined intervals follows. While it gives rise to latency, reliable data processing is guaranteed, and only once. And, obviously, Spark Streaming interfaces flawlessly with conventional Spark processing, allowing easier development.

Flink is merely a Spark and Storm hybrid. While Spark is a batch framework with no true streaming support and Storm is a streaming framework with no batch maintenance, Flink streaming and batch processing capability structures. It enables low latency offering of Storm along with Spark's data fault tolerance, along with numerous user-configurable windowing and redundancy settings. Lack of prevailing production deployment, and lack of inherent commercial assistance from major Hadoop distributions, is a drawback of Flink.

2.8 What does the future hold?

Evolving and expanding big data environment holds the future of data lakes.

The Hadoop ecosystem is realizing extraordinary approval and it being an assembly of open source ventures backed by the community implies that progress and growth happens at a greater rate than that of conventional softwares.

Produces like NiFi, Ignite Streams, Beam, Samza and numerous potential advancements are on a boom. Ingestion, and specifically streaming ingestion, is time consuming and an intricate process that can converge at a twisted state at an instant. Tools such as Zaloni Bedrock allow association of distinct solutions, regardless of the selection made for business, even as it grow, scale and evolve the Hadoop ecosystem.

3. Conclusion

Data Lake is an outcome of burning need to administer and manipulate new data variants. In substance, the current requirements will mold the data lake regardless of the prevailing data processing framework. Only experimentation can lead to the appropriate data lake establishment.

The data lake and the enterprise data warehouse in collaboration results in a cooperation of competences that delivers fast-tracking returns. Permitting people to work with data quicker and motivating business outcomes: That's the dominant outcome of investing in a data lake to add to the enterprise data warehouse

References

[1] PwC, "Data lakes and the promise of unsiloed data," pwc.com, May 6 2015. [Online]. Available: http://usblogs.pwc.com/emerging-technology/data-lakes-and-the-promise-of-unsiloed-data. [Accessed: June 10, 2017]

[2] Carlos Maroto, "A Data Lake Architecture With Hadoop and Open Source Search Engines," dzone.com, April 4, 2016. [Online]. Available: https://dzone.com/articles/a-data-lake-architecture-with-hadoop-and-open-sour. [Accessed: June 11, 2017]

[3] Tamara Dull, "Data Lake vs Data Warehouse: Key Differences," kdnuggets.com, [Online]. Available: http://www.kdnuggets.com/2015/09/data-lake-vs-data-warehouse-key-differences.html. [Accessed: June 15, 2017]

[4] Greg Wood, "Top Streaming Technologies for Data Lakes and Real-Time Data," zaloni.com, September 20, 2016. [Online]. Available: https://resources.zaloni.com/blog/top-streaming-technologies-for-data-lakes-and-real-time-data. [Accessed: June 20, 2017].

Author Profile

Kartikay Sharma is a final year student of B.Tech in Computer Science at JRE Group of Institutions, under Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh. He has a knack for playing with data at hand and aspires to become a Data Scientist.

Shreya Patoa is a senior year student at JRE Group of Institutions, affiliated to Dr. A.P.J.Abdul Kalam University. She takes interest in advancing telecommunication technology and aims to soar with it.