A Comparative Study of Recent Trends in Big Data

Pwint Phyu Khine[1], San Myint Tin[2], Soe Mya Mya Aye[3]

Abstract

It was estimated that there will be 181 zeta bytes of data in 2025 terming big data. The unexpected occurrence of Covid-19 makes data volume consumed skyrocket making understanding and manipulating such an amount of big data to extract valuable information become a necessary challenge. Data becomes new oil. Challenges for these big data make great changes in the data landscape leading to recent trends in big data. The main prominent trends are the ideology of polyglot persistence to use different data stores with different characteristics in single application, the revisiting of data warehouse concepts and emergence of data, and the choice of machine learning algorithms or ML algorithm to be revisited due to Big Data V characteristics.

Keywords: Big data, Big data characteristics, NoSQL data stores, polyglot persistence, Data Lake

1. Introduction

Big data refers to data that cannot be handled by traditional tools and techniques. Google handles Petabytes of data including user queries, search results, etc., generating and consuming an unprecedented amount of data. Facebook has over 2 billion users. Amazon offers services all over the world. The widespread use of mobile devices, social media, the Internet of Things (IoT), and the advances in artificial intelligence (AI) generate and consume data more than ever, creating big data everywhere. Big data becomes a necessity to apply in multidisciplinary fields. During the Covid-19 outbreak, the Pzifier company applied machine learning with big data to study the possible effects of chemical compounds for producing Pzifier. The health care industry has widespread use of big data from public health care to the discovery of new cures (Dash, S., Shakyawar, S.K., Sharma, M. et al. 2019)

Not only does data volume increase, but also types of data differ ranging from traditional structured data to current images, voices, videos, and, finally, holograms and virtual spaces. Facebook virtual space is one of such features in big data technology landscapes. Data from the Hubble telescope are gathered and visualized, making it possible to uncover more secrets of the universe in the big data era. The complex lifecycle of data systems makes the data differ in speed. These lead to prominent big data characteristics namely V's in big data – volume, velocity, and variety (Doug Laney,2001).

The abundance of data overwhelmed become challenges to be able to retrieve valuable information, apply that information to gain knowledge, and in turn, achieve intelligence (computer science field) or business insights/business intelligence (business field). When facing these challenges, researchers have to make the traditional approaches more sophisticated or have to develop newly established concepts, or completely abandon the previously accepted theories. Sometimes, it is more of a tradeoff between newly established concepts and traditional concepts (Al-Jaroodi, J., & Mohamed, 2017).

The objective of the present study is first to observe traditional concepts related to current trends in today's data world, then, to introduce the newly established or renaissance concepts, and to make a comparison of how they are being applied in the big data age.

1 Demonstrator, Department of Computer Studies, University of Yangon.
2 Demonstrator, Department of Computer Studies, University of Yangon.
3 Professor (Head), Department of Computer Studies, University of Yangon.

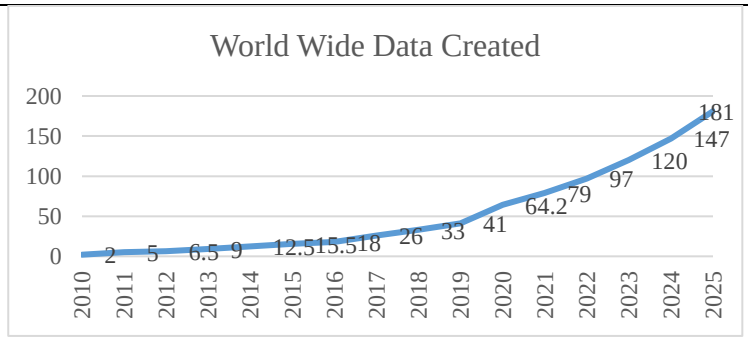
2. Big Data and its Characteristics

Big data has defining characteristics, which are defined in Vs – V characteristics. There are different suggestions for V characteristics. However, the most widely accepted V characteristics are volume, variety, and velocity.

2.1 Volume

Volume: A large amount of data is generated or consumed by today's data landscapes. Social media alone generate a tremendous amount of data such as user data, texts, images, audio, and videos. Figure 1 shows the total amount of data consumed globally increases rapidly, reaching 64.2 zettabytes in 2021. Prediction for next coming years up to 2025 is shown in figure 2. The COVID-19 pandemic unintentionally created more people to engage for work from home, making data created reach a new high.

Bits	0 or 1
Kilobyte (KB)	8 bits (e.g. 000 000 01 = 1)
Megabyte (MB)	1024 Kilobytes (10^3)
Gigabyte (GB)	1024 Megabytes (10^3)
Terabyte (TB)	1024 Gigabytes (10^3)
Petabyte (PB)	1024 Terabytes (10^3)
Exabyte (EB)	1024 Petabytes (10^3)
Zettabyte (ZB)	1024 Exabytes (10^3)
Yottabyte (YB)	1024 Zettabytes (10^3)



https://www.statista.com/statistics/871513/worldwide-data-created/

Figure 1. Data Measure and the growing big data predicted to 2025

2.2 Variety

Variety: Data can further be divided into structured. semi-structured and unstructured data. Structured data include traditional data transactional from file systems which are formatted such as relational databases using Codd's relational theory with Entity Relationship Diagrams as shown in figure 3. Semi-structured data includes XML (eXtensible Makeup Language), JSON (JavaScript Object Notation), CSV (Comma-Separated-Value) etc. An example of semi-structured data is shown in figure 4. Unstructured data include texts (i.e. with different meanings and grammar), images, audio, and video data.

Example of Structured Data in SPSS

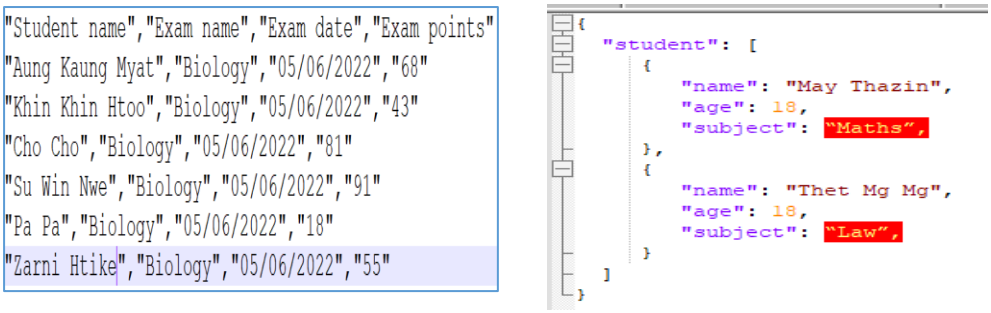
Name	Type	Wi...	Decim...	Label	V...	M...	...	Align	Meas...	Role
V1	Numeric	3	0	Number	N...	N...	8	Right	Scale	Input
Sepal.Length	Numeric	3	1	Sepal Length	N...	N...	8	Right	Scale	Input
Sepal.Width	Numeric	3	1	Sepal Width	N...	N...	8	Right	Scale	Input
Petal.Length	Numeric	3	1	Petal Length	N...	N...	8	Right	Scale	Input
Petal.Width	Numeric	3	1	Petal Width	N...	N...	8	Right	Scale	Input
Species	String	10	0	Species	N...	N...	10	Left	Nom...	Input

(a) Design View

	V1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Sp
1	1	5.1	3.5	1.4	.2	setosa
2	2	4.9	3.0	1.4	.2	setosa
3	3	4.7	3.2	1.3	.2	setosa
4	4	4.6	3.1	1.5	.2	setosa
5	5	5.0	3.6	1.4	.2	setosa
6	6	5.4	3.9	1.7	.4	setosa
7	7	4.6	3.4	1.4	.3	setosa
8	8	5.0	3.4	1.5	.2	setosa
9	9	4.4	2.9	1.4	.2	setosa
10	10	4.9	3.1	1.5	.1	setosa
11	11	5.4	3.7	1.5	.2	setosa
12	12	4.8	3.4	1.6	.2	setosa
13	13	4.8	3.0	1.4	.1	setosa
14	14	4.3	3.0	1.1	.1	setosa

(b) Data View

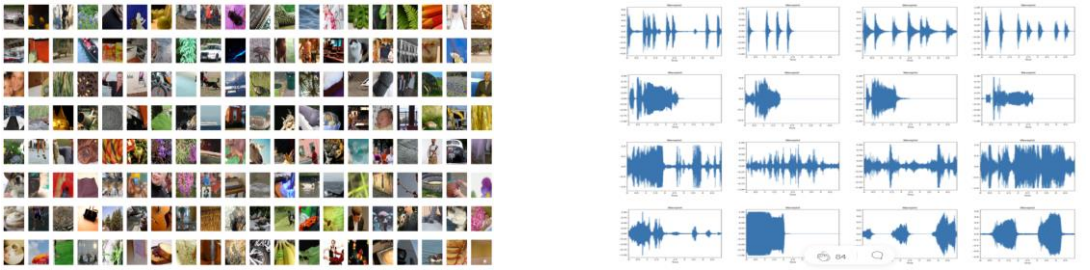
Figure 2. Examples of Structured Data in SPSS



(a) Semi-structured data with .csv extension (Comma-Separated-Value)

(b) Semi-structured data with .json extension *Note: JSON has types and structures defined subtly*

Figure 3. Example of Unstructured Data



(a) Image dataset example (Unstructured)

(b) Audio dataset example (Unstructur

Figure 4. Comparison of Structured, Semi-Structured And Unstructured Data

2.3 Velocity

Velocity: Data speed can be further divided into batch, near real-time, real-time, and streaming. There is also CEP, i.e., complex event processing that will only prompt when there is a significant trigger. A system could be built based on the requirement –e.g. scholarship applications system could work in batches. Advertisements for mobile phone are working in near real-time by monitoring user browse histories and other interests. Insulin monitoring or heartbeat monitoring system will work in real-time. Lives sales will work on streaming. Some countries provide better government and private businesses by progressing from batch systems into near real-time systems such as modifying into one-stop online visa applications, mobile electronic billings for businesses and personal usage, etc. This also increases the velocity aspect of the big data for big data system implementations.

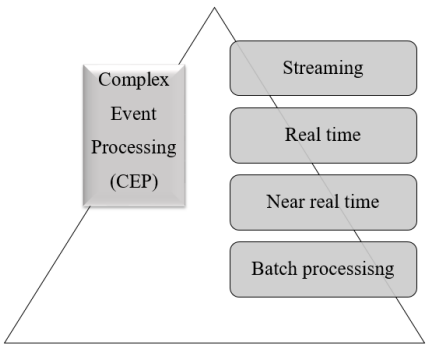


Figure 5. Big Data Velocity Requirements

Another V's characteristics that are also accepted are Variety – which ensures the data quality, Value for gaining insight from data, and Variability for nuances of data changes

brought by velocity characteristics. A custom saying in computer science is "Garbage in Garbage out". If input data is not ensured of quality, output results will be the same poor quality. The value will be harder to extract in the case of such scenario.

3. Different Data Stores and Polyglot Persistence

In big data era, data bases are sometimes called as data stores interchangeably. The prominent databases i.e. data stores that dominated the database world are relational databases based on Codd's relational theory and set theory such as MySQL, Microsoft SQL, Oracle, Access, etc. They are based on SQL (structured query language) for data manipulations and are given the name, SQL data stores. All data have to be formatted to be in the form of relations i.e. tables. ER (Entity Relationship) is used to conceptualize the relationships between these tables (relations). They are diagrammed into ER Diagrams. SQL databases are row-oriented for data retrieval. The basis for every data stores technologies – is CRUD operations, namely C – Create, R – Retrieve, U – Update, and D – Delete. These operations are carried out using the standardized structured query language (SQL). They are based on atomic consistency level i.e. all or nothing. If one operation is done, it is done. Else they won't be executed at all, featuring that the characteristics of SQL as ACID – Atomicity, Consistency, Isolation, Durability.

However, Big Data brought a variety of aspects making SQL databases rigid and difficult to control. Many relational databases, i.e., SQL data bases work well with ACID characteristics, i.e. atomicity, consistency, isolation, and durability. Because of the volume aspect of big data, distributed and parallel computing become a necessity for big data making it difficult to fully applicable ACID in all data stores which are not based on relational models. This led to the revisiting of the CAP (consistency, availability, partition tolerance i.e. durability) theorem (Brewer 2002), leading to the acceptance of BASE characteristics in some data stores that are not SQL. These data stores are called "Not Only SQL" because some of them can offer ACID characteristics. (Banothu, N., Bhukya, S., & Sharma, K. V. 2016.)

3.1 Comparison of SQL and Newly Emerged NoSQLs Data Stores

Different from ACID, BASE characteristic only requires data stores to be basically available, soft state, and eventual consistency proving that one of the CAP properties will sometimes need to be compromised if the distributed file system is oriented towards availability properties (Brewer, E. A.,2012). If the work of data stores i.e. NoSQL data stores are in line with such BASE characteristics, they are considered acceptable data stores in big data because they have to work with distributed and parallel computing. (Abadi, D. 2012.) Most prominent NoSQL data stores are key-value data (e.g. Raik,) columnar data stores (e.g. Cassandra), document data stores (e.g. MongoDB), and graph data stores (e.g. Neo4j). The comparison analysis for SQL and NoSQL data stores is shown in Table 1.

Some of the researchers extended or adopted other data stores from different fields such as time-series, spatial, temporal, and spatial-temporal data stores, etc., as "Not Only SQL" data stores. However, the four most widely accepted NoSQL data stores are

i. Key-value stores: They are based on simplified keys and their associated values. Values are retrieved and manipulated using keys.
ii. Columnar stores: They are column-oriented in their data storage and retrieval, unlike relational SQL data stores which are row-oriented in data manipulation. By doing so, they eliminated the null value problems which waste a lot of valuable memory space in SQL data stores.
iii. Document stores: They are the more sophisticated version of key-value stores. Their values are documents that need to be manipulated. Most of them could not

incorporate the full ACID characteristics in their nature, they are implemented to follow at least eventual consistency.

iv. Graph data stores: Unlike other NoSQL data stores which compromised with BASE characteristics, they mostly accept the ACID principles. Entities are represented as nodes, their relationships are represented by edges, and the entity (and relationships) values are considered as properties. Not only entities but also relationships may have the properties.

Table 1. Comparison Analysis of SQL and NoSQL Data Stores

Features	SQL Data Stores	NoSQL Data Stores
Model	▪ Codd' Relational Model ▪ Rows-Oriented ▪ Entity Relationship (ER)	Differ based on the types of the data stores and vendors i. Key-value stores/ Document stores (Simple Keys-Values) ii. Columnar – Columns Oriented iii. Graph – Graph Theory
Characteristic	ACID	BASE (mostly)
Key	Primary Keys, Foreign Keys	i. Keys and values pairs (Sometimes nested) ii. Columnar – Column Family iii. Graph – Nodes
Query	Structured Query Language (SQL)	Depends on the types of data stores i. Key-value stores/ Document stores (Values Retrieval based on Keys) ii. Columnar – Query Language similar to SQL iii. Graph – Cypher Query Language

3.2 Polyglot Persistence

Variety aspects of big data bring out using different data stores i.e. SQLs and NoSQLs in a single application system. A person who can speak many languages is called a polyglot. In computer science, different programming languages in a single application are called polyglot processing (Michael Hausenblas,2014).

Martin Fowler recoiled the term Polyglot Persistence using different kinds of conceptually different data stores for the single systems. "Polyglot persistence, i.e. using different databases appropriate for different parts in one system, is becoming prevalent in data-intensive systems of big data as they are distributed and parallel in nature." (Khine, P.P.; Wang, Z.2019). There is no "one size fits all" data stores to fulfill the big data characteristics although database vendors of both SQL and NoSQL try to provide more properties to their stores aiming to become "one size fits all" solution. There are many investigations about polyglot persistence for implementation in healthcare. Some exemplar systems are implemented applying the concepts of polyglot persistence (Prasad, S., & Avinash, S. B.,2015.) They use different kinds of SQL and NoSQL data stores for a particular service.

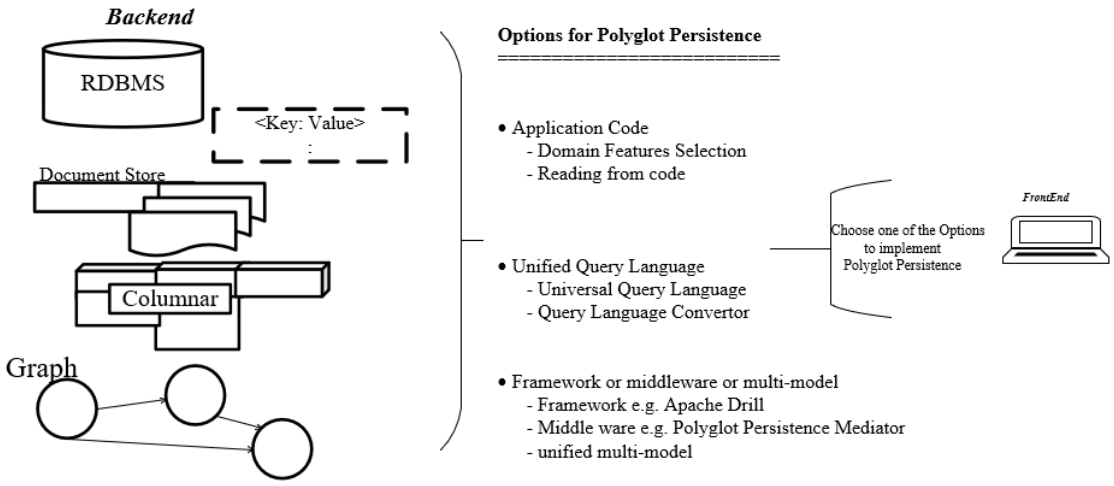


Figure 6. Polyglot Persistence (Khine, P.P.; Wang, Z.2019)

Polyglot Persistence could be achieved in three ways. The first is by performing feature selection – performing feature engineering when necessary. Then, application code will manipulate these data features and perform necessary engineering. The second is the advancement in the development of query languages. For specific data store e.g. Neo4j graph data stores use the Cypher Query Language and its developers are adding more features in their data stores capability. Some are trying to develop universal query language for different data stores. Query Language Convertors are also possible to provide polyglot persistence. The third possible implementation is the use of framework or middle or multi-model that aims to provide polyglot persistence.

3.3. Data Repositories – Comparing Data Warehouse And Data Lake

A comparative analysis of traditional data warehouse and the newly emerged data lake is performed in this section.

3.3.1 Data Warehouse

Definitions of data warehouse – subject-oriented, integrated, time-variant, and non-volatile. They relied on traditional extract. Transform, load (ETL). Data from structured data sources are extracted. In a data warehouse, required data from an operational system (mostly relational data stores) are first extracted. These data are further preprocessed. Data cleaning is performed to fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies. Data from multiple databases (mostly relational databases), and files are integrated, and sometimes, data cubes are created or integrated. Data reduction such as dimensionality reduction techniques is applied. Sometimes, data is compressed for reducing its size. Finally, data transformation is performed by normalizing data and generating the necessary concept hierarchy. These highly processed data are finally loaded into the data warehouse to provide Online Analytical Processing - OLAP, reporting for business processes, data mining, and so on. Sometimes, data warehouses implement data cubes for faster query retrieval. Data warehouses are highly structured and worked well with SQL relational data stores. However, data variety makes new challenges to them. This alone led to the revisiting of data warehouse concepts and the creation of a data lake. Sometimes, data cubes are built in the data warehouse to better provide better query retrieval for analytical processes. Metadata – information about data is used for the manipulation of data warehouses. They are also considered schema on write approach (i.e. creating a schema for data before writing).

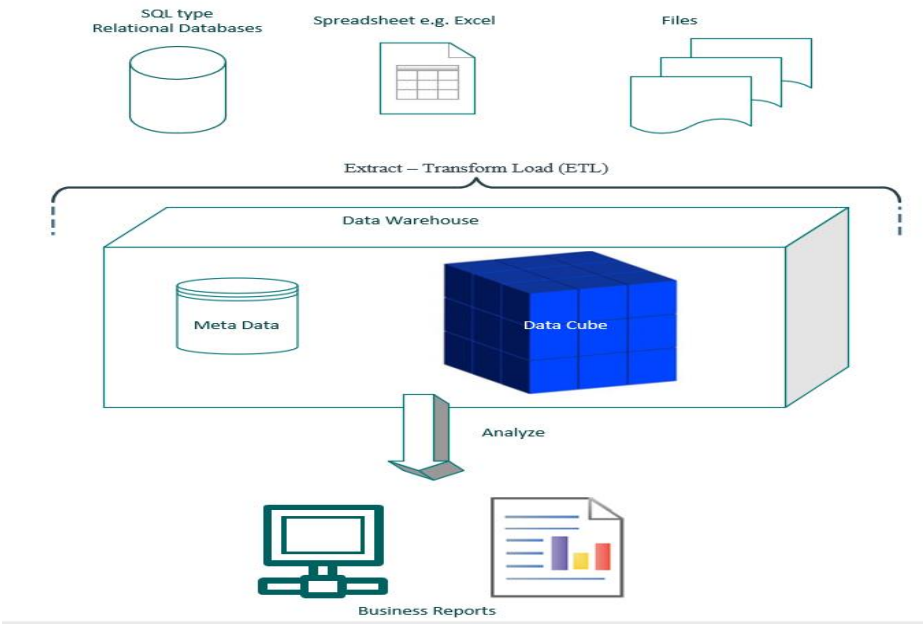


Figure 7. Simplified Data warehouse architecture

3.3.2 Data Lake

In Data Lake, conceptually, data should be stored in native format. Because of the variety aspect of the big data characteristic, data could come from many different sources and in many forms which may or may not be convenient in relational form. Data velocity could be different such as batch processing for Netflix recommendation engines to real-time sensing of air pollution from IoT devices. Therefore, data in Data Lake should be stored in their native formats. Instead of the ETL process in the data warehouse, Data Lake changes the data pipeline process into the Extract, Load, and Transform – ELT process. All data are extracted from their original system (Khine, P. P., & Wang, Z. S, 2018). They are also considered schema on read approach i.e. schema creation may be ad hoc for data writing,

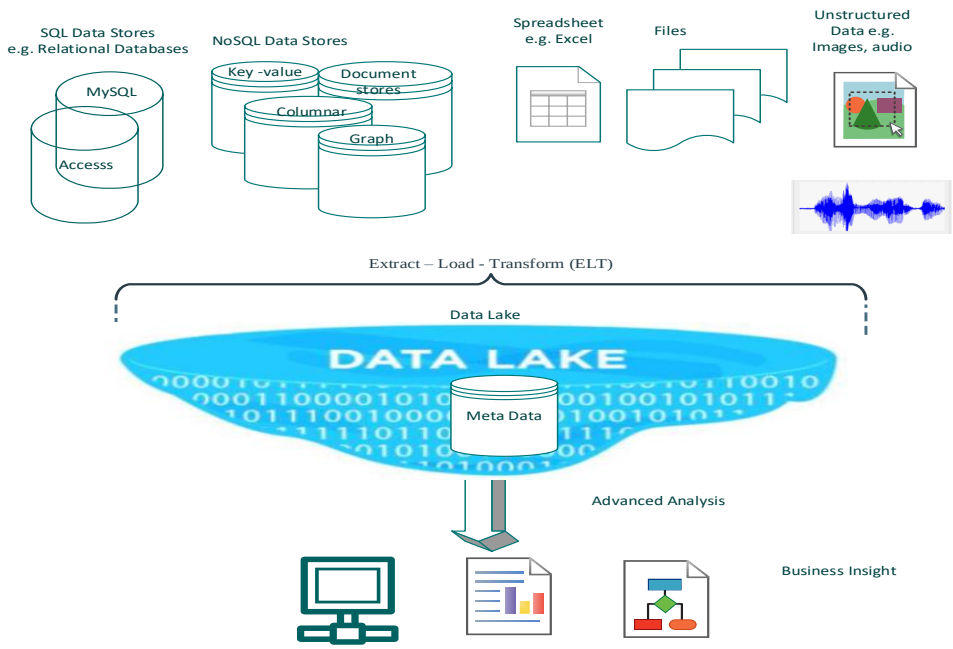


Figure 8. Simplified Data Lake Architecture

Both data lake and data warehouse have to rely on metadata. Metadata is classified into two categories intra-metadata (i.e. the characteristics, meaning, quality, and security level of datasets can be analyzed and understood by users), and inter-metadata – (to explore data based on the requirements of the user's relevant dataset). Data characteristics are mainly concerned with a general overview of information such as dataset names, date created, size, dataset origin, etc. Data lineage and data profiling are used to be able to trace back the data in its original stage. Definitional metadata specifies dataset meaning. It is usually classified into semantic metadata and schematic metadata. If the dataset contains unstructured data (most of the time a mixture of structured data), it is semantically represented by some keywords or texts. Dataset locations are represented by file paths and URLs of the database connections called navigational metadata (Prasad, S., & Sha, M. S. N. 2013.).

A poorly managed data lake can become a data swamp that is not only poor in structures but also makes data typically difficult to retrieve. In this ideology, a data lake should be able to collect all kinds in its native form. However, data with different conceptual characteristics for consistency, differing data velocity, data linage, poorly recorded metadata, undefined queries, and data profiling difficulties can lead to a dysfunctional data collection mess.

Some of the big companies try to extend their data warehouse. They named their big data warehouse. They try to provide both data lake and data warehouse capability at relatively high however most of them are proprietary. In reality, both data warehouses and data lakes are finding suitable grounds, and businesses are using them both based on their needs. A comparison study of data warehouse and data lake is shown in table 2.

Table 2. Comparison of Data Warehouse and Data Lake

Features	Data Warehouse	Data Lake
Data	Structured	Structured, Semi-Structured or Unstructured
Process	Extract-Transform-Load (ETL)	Extract-Load-Transform (ELT)
Schema	Schema-on-Read	Schema-on-write
Mode	Batch	Batch, try to support different data velocity
Sources	SQL Data Stores (relational), Spreadsheets, limited types of files, Try to provide NoSQL data stores which have definitive structure	Try to provide both SQL (relational) and NoSQL stores (key-value, documents, columnar, graph) and data with undefined structures

3.4. Machine Learning In Big Data

Traditional machine learning models are based on a single computer with very few samples. In big data era, modifying traditional ML algorithms is oriented towards scalability. before the era of big data, machine learning capabilities are overestimated and could not scale out, resulting in one of the infamous AI winter. Today, machine learning algorithms apply the divide and conquer strategy. One of the successes of the machine learning models using big data is area of unstructured data such as image processing, audio and video processing. Previously, it was impossible to train such a big amount of data and inherent complexity. However, now it is becoming possible because of the advances in data technologies especially by modifying ML algorithms. ML has great impacts on a wide variety of applications ranging from simple Machine learning (ML) techniques that have generated huge societal impacts in a wide range of applications such as NLP (natural language processing), computer vision, audio and video processing, Internet of Things, etc.

ML algorithms are revisited based on the V's characteristics of big data (i.e. volume, variety, and velocity). ML is oriented to provide one or all of the characteristics by training different types of structured and unstructured data. The basic principle of the MapReduce model (Jeffrey Dean and Sanjay Ghemawat,2008) comes from Divide and Conquer Strategy from Data structure and algorithms. Examples of traditional decision tree algorithm and random forest which is more suitable for big data are visualized. Due to limited time, only sample data is used to perform the following visualization for comparative study. The dummy dataset is the marks of different subjects the first year CS student got in their exam. There are two classes – pass and fail. Decision tree algorithm and random forest algorithm use the same dummy dataset to visualize their inner working.

```
Train_data = data[0:20,:]
#Train_data
Train_target = target[0:20]
#Train_target

Test_data = data[20:,:]
#Test_data
Test_target = target[20:]
#Test_target

decisionTree_model =  tree.DecisionTreeClassifier()
decisionTree_model.fit(Train_data,Train_target)
predict_target = decisionTree_model.predict(Test_data)
tree.plot_tree(decisionTree_model.fit(Train_data,Train_target))
```

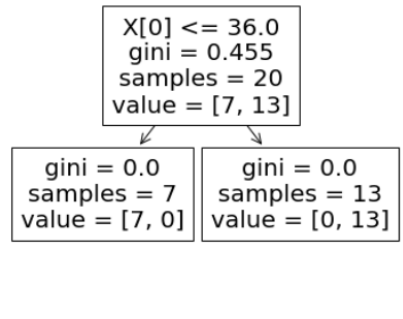


Figure 10. Visualization of how Simple Decision Tree Algorithm works

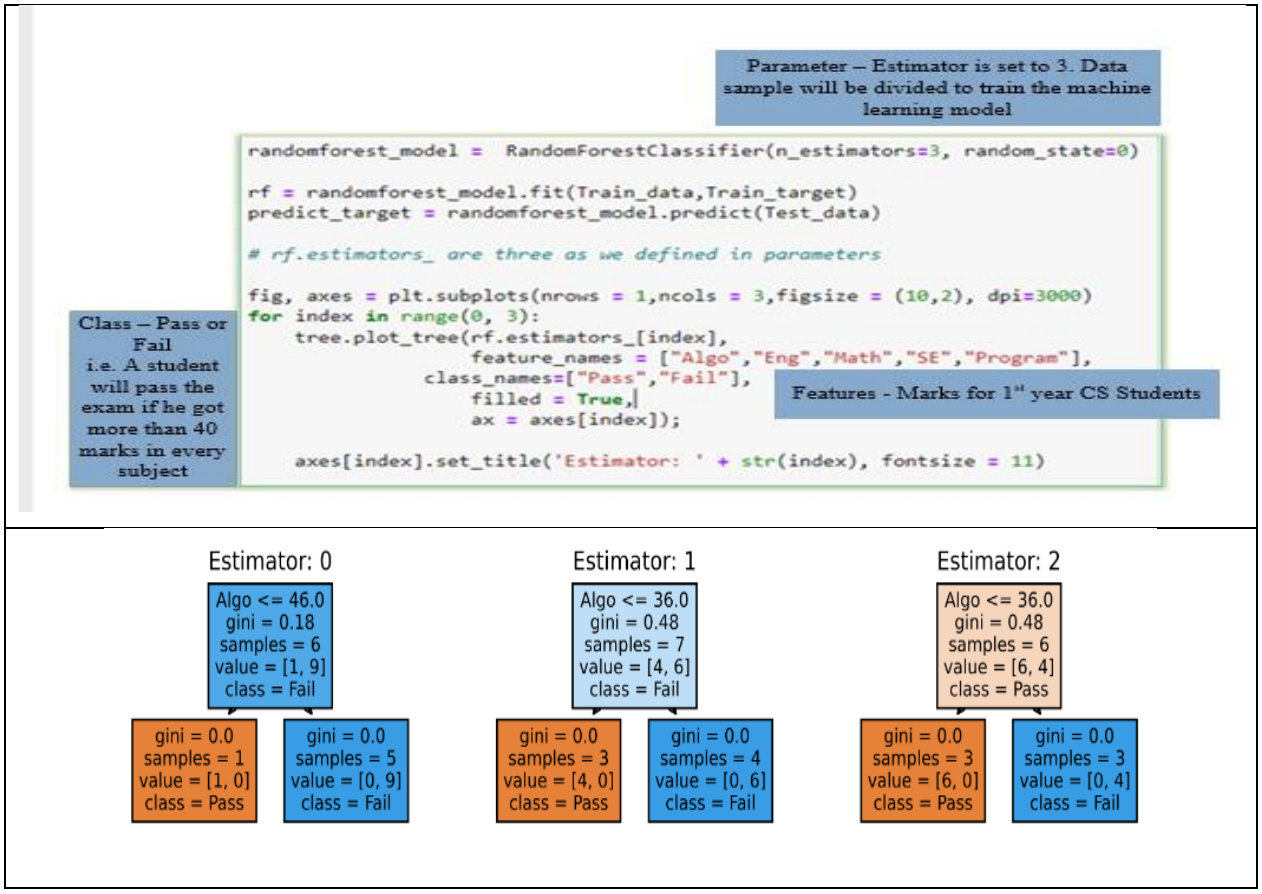


Figure 11. Visualization of how Random Forest Algorithm works

Machine learning can mainly be grouped into three categories: supervised – train the data knowing class labels, unsupervised –for unknown class labels, and reinforcement learning – the system rewards or get a penalty based on performance. Deep Learning (DL) is another subset of ML which found its niche in unstructured dataset such as image, audio and video processing. Traditional decision tree algorithms are performed in memory. These may sometimes lead to the problem when the data volume becomes very large. Random Forest Algorithm is a variety of decision tree algorithms which is suitable for big data by diving data and computing resources.

Machine learning algorithms for big data solve the complexity and memory limitation problem using distributed computing by allocating learning processes to multiple computers and performing data parallelism and model/parameter parallelism. (Dash et al., 2019) Data parallelism means leveraging existing big data architecture by partitioning the data and computing partitioned data. Model (parameter) parallelism creates parallelized versions of learning algorithms. Learning model/parameters are divided, and each structural block is concurrently computed to unleash the hidden big data value.

A survey of ML algorithms for big data is already conducted in the work of Rajkumar (Rajkumar Buyya, Rodrigo N Calheiros, Amir Vahid Dastjerdi, 2015). Therefore, only an example of algorithm nature visualization is performed in this study. Machine learning requires training with labeled data as shown in example, randomforest is preferred in big data ML over traditional simple decision tree algorithms because it is more suitable for distributed and parallel computing to solve scalability issues. Another choice is decision tree algorithms which will be revisited to develop new kinds of ML algorithms that are more suitable for handling big data (Jiawei Han, Micheline Kamber, Jian Pei, 2008). Big data make the Machine Learning algorithms possible to extract underlying data patterns that have been unable to discover.

4. Discussion and Conclusion

Big data become inevitable in the data world. Every day people generate big data and consume big data consciously or unconsciously. In this study, big data characteristics are discussed such as where they come from, and their applications. Among them, V's characteristics of big data are prominent. Different data stores based on big data are also discussed. From structured, semi-structured, and unstructured data, different data stores become applicable in the big data era. They make truly evolutional concepts such as revisiting ACID characteristics to BASE and rethinking traditional atomic consistency with eventual consistency. Newly established concepts such as Polyglot Persistence become prominent. Traditional data repository concepts such as data warehouses have to be revisited, and data lake concepts such as storing data in native formats become to emerge for exploring more data to extract more value. Machine learning algorithms impacted by big data characteristics are also discussed pointing out the revisiting of old concepts or the emergence of new ideologies.

Acknowledgment

The author would like to give my sincere thanks to Dr. Soe Mya Mya Aye, Professor, Head of Department of Computer Studies, University of Yangon, for her helpful pointers in improving this work. I also paid my gratitude to Dr. Wint PaPa Kyaw, Professor, for helping me find necessary reference works. I also thank Dr. Thet Thet Hlaing, Professor, Dr.Khin Sandar Myint, Associate Professor, and Daw Su Thandar Aung, Lecturer, Department of Computer Studies, University of Yangon for their guidance in all matters. Finally, I gave my special thanks to coauthor Daw San Myint Tin in for her active cooperation in doing this work.

References

Abadi, D. (2012) **Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story.** Computer, 45(2), 37–42. doi:10.1109/mc.2012.33

Al-Jaroodi, J., & Mohamed, N., (2017). **Characteristics and requirements of big data analytics applications.** Proceedings - 2016 IEEE 2nd International Conference on Collaboration and Internet Computing, IEEE CIC 2016, 426–432. https://doi.org/10.1109/CIC.2016.062

Brewer, E. A. **(**2000) . **Towards robust distributed systems (abstract). Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing** – PODC 00. doi:10.1 145/343477.343502

Brewer, E. A., (2012) **. CAP twelve years later: How the "rules" have changed**,". Computer vol. 45, no. 2, pp. 23-29. doi: 10.1109/MC.2012.37

Both, N., Bhukya, S., & Sharma, K. V. (2016) **Big data: Acid versus base for database transactions. International Conference on Electrical, Electronics, and Optimization Techniques**, ICEEOT 2016, 3704–3709. https://doi.org/10.1109/ICEEOT.2016.7755401

Doug Laney. **3D Data Management: Controlling Data Volume, Velocity, and Variety**. Technical report, META Group, Inc (now Gartner, Inc.), February. 2001. [Online]. Available http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

Dash et al., (2019) , **Big data in healthcare: management, analysis, and future prospects, J Big Data** . https://doi.org/10.1186/s40537-019-0217-02019,

Jeffrey Dean and Sanjay Ghemawat, **MapReduce: Simplified Processing on large clusters** , Communication of the ACM, Vol. 51, No. 1, Jan 2008.

Jiawei Han, Micheline Kamber, Jian Pei, **(** 2008), **Data mining: concepts and techniques, 3rd Edition**

Khine, P. P., & Wang, Z. S. (2018). **Data lake: a new ideology in big data era. ITM Web of Conferences, 4th Annual International Conference on Wireless Communication and Sensor Network (WCSN 2017),** Volume 17, 03025. https://doi.org/10.1051/itmconf/20181703025

Khine, P.P.; Wang, Z.(2019) . **A Review of Polyglot Persistence in the Big Data World.** Information 2019, 10, 141. https://doi.org/10.3390/info10040141

Michael Hausenblas, **Polyglot Processing**, July 6, 2014 [Online]. Available: http:// datadventures. ghost.io/2014/07/06/polyglot-processing

Pramod J. Sadalage, Martin Fowler, (2012), **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence** [1 ed.]

Prasad, S., & Sha, M. S. N. (2013). **Next generation data persistence pattern in healthcare: Polyglot persistence. 2013 4th International Conference on Computing, Communications and Networking Technologies**, ICCCNT 2013. https://doi.org/10.1109/ICCCNT.2013.6726734

Prasad, S., & Avinash, S. B. (2015)**. Application of polyglot persistence to enhance performance of the energy data management systems. 2014 International Conference on Advances in Electronics, Computers and Communications,** ICAECC 2014 https://doi.org/10.1109/ ICAECC.2014.7002444

Rajkumar Buyya, Rodrigo N Calheiros, Amir Vahid Dastjerdi, (2015), **Big Data Principles and Paradigms** , Elsevier, Cambridge, USA