UNIVERSITY OF
LINCOLN

Adewusi Adedapo Adetomiwa

29487352

School of Computer Science

29487352@students.lincoln.ac.uk

Big Data Analysis and Modelling

Module Co-ordinator: Dr Miao Yu

March 13th, 2025

Section I

A distributed big data processing ecosystem is designed to analyse large volumes of data, also known as big data, across multiple machines or nodes. The system comprises tools and technologies that help to process and store data, which serves to store and analyse data effectively and efficiently. Systems like Hadoop Distributed File System (HDFS) have helped store data across multiple nodes and monitors from various sources. Frameworks like Apache Spark have helped in processing the data across the various nodes with speed while collecting and transporting data into the ecosystem from different sources with the help of data ingestion tools like Apache Kafka or Flume, which ingest data in batches or in real-time and designed to be fault tolerance and scalable.

Daily data from many sources is increasing dramatically in the modern era (Cappa et al., 2021). Nonetheless, big data has made a significant contribution and is a vital asset for organizations all over the world (Ferrigno et al.,2023).

Describing the ecosystem with big data was further described in a literary work as a collection of intricate datasets in their volumes, encompassing real-time data, data management capabilities, and enormous amounts of data (Anuradha, J. 2015). Big data, according to Gurjit et al. (2019), It's a vast compilation of historical and significant data, serving as an organization's most precious asset, leveraged effectively for informed business decisions instead of relying on gut feelings. Nevertheless, big data are more significant than small data, requiring different techniques, tools, and architecture.

Big data comprises five characteristics: Value, Velocity, Variety, Volume and Veracity.

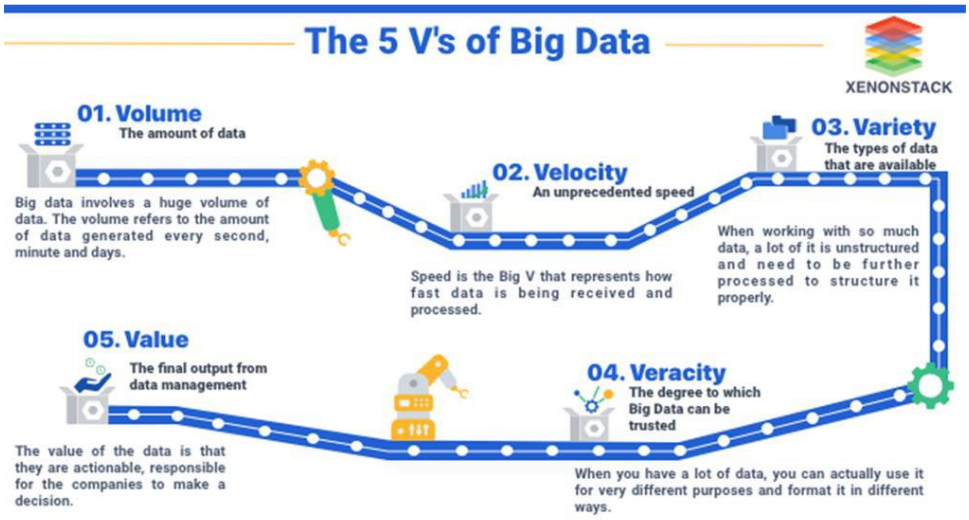


Figure 1: Big 5V of Data gotten from C. Gaur (Online)

The data size that needs processing is increasing rapidly, and storage is also increasing. More computation tools and techniques are also being applied in a distributed ecosystem. Dautov et al. (2017) defines big data as an enormous amount of data generated per second from many sources. It has been proven that the impact of using Hadoop and Spark to deal with large volumes of data in the distributed ecosystem cannot be overemphasised (Dasgupta et al., 2018).

The Velocity of every distributed big data system represents how fast data is being created, received from different sources, and processed using Apache Kafka. Flume helps with the real-time data stream (Anuradha, 2015). Data is generated at the speed of millions of messages being transferred at any given time across social networks (Dasgupta et al., 2018). According to research, 900 million photos are uploaded daily to Facebook, and 3.5 billion searches happen on Google daily.

Variety

The big data ecosystem includes various formats of data type and structure, such as text, numerical, images, audio, etc. Big data can also be in different forms, such as textual, sensor, media, or streaming (Dasgupta et al., 2018). This is illustrated below.

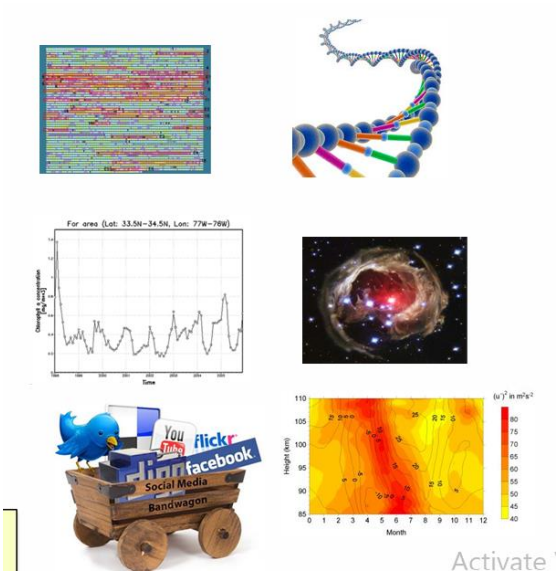


Figure 2: Variety in Big Data Analytics Gotten from Lecture note

Veracity

This is the Noise inherent in data, such as inconsistencies in recorded information that require validation (Dasgupta et al., 2018). The messiness or trustworthiness of data, for instance, Twitter posts with different hashtags, can be messy, but big data technology helps to deal with it.

Value

Value is the most important V in Big Characteristics. Big data has immense promise, but it needs to be well-maintained and easily accessible (Anuradha (2015). According to Anuradha et al. (2015), value is the result of processing big data, both the output and insight gained from it.

Fault Tolerance and Resilience

This ability of data that is used in distributed big data processing to maintain availability, reliability, and constant performance is Fault tolerance (Saadoon et al., 2022). Fault tolerance, which effectively prevents the system from losing data through a redundant system, is defined as the failure of one or more processing nodes without affecting the proper operation of the entire system. This is a significant challenge for stream processing systems, according to a further review of Zhao's (2017) literature. It can also mean that a system can restart the failed node to restore its functionality and replicate important data before a failure occurs. The figure below was used to illustrate this further.

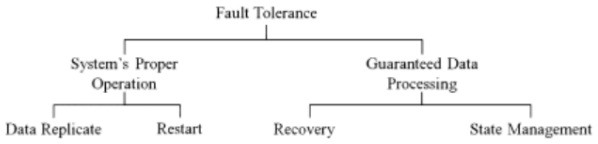


Figure 3: Fault Tolerance in a Distributed System (Zhao, 2017)

Evaluating the technique effectiveness of distributed Big Data in an ecosystem reveals a need for Resilience strengthening, which enables the system to withstand and recover from failure and minimise downtime and data loss (El Baqqaly et al., 2023). Data resilience is the ability or capability to withstand shock and disruption (Raut et al., 2021).

Distributed File System (HDFS)

This is always processed using Hadoop, a framework for managing big data storage and processing it parallelly. Hadoop is written in Java and consists of HDFS, the storage unit, and MapReduce, the processing Unit. HDFS contains two nodes, Name and Data nodes.

Apache Hadoop is a freely available framework that facilitates the scalable and decentralized handling of large-scale data across networked server groups, employing uncomplicated coding models to manage tasks efficiently (Anuradha et al., 2015). HDFS configuration is based on enslaver and enslaved person patterns, making the NameNode a controller node and all DataNodes an agent node (Voran. D. 2018). In the ecosystem, the NameNode stores the metadata and nothing else and keeps track of all data sent to DataNodes, while the DN reads the data, writing and processing the data.

Resource Manager and Scheduler

These are vital components in a distributed big data ecosystem, ensuring efficient utilisation of computational resources and timely processing of large datasets. These algorithms improve big data performance, ensuring that resources are used optimally and tasks are completed within the timeframes. The main role of the resource manager is to keep track of the resources across the cluster and schedule applications. It mainly consists of two components as scheduler and an applications manager (Voran D., 2018). YARN (Yet Another Resource Negotiator)

According to Voran, 2018, YARN distinguishes the resource management, scheduling and processing components from one another to achieve 100% resource efficiency of the cluster in an ecosystem. YARN (Yet Another Resource Negotiator) coordinates compute resources such as processing power and memory allocation within Hadoop clusters, guided by configurable scheduler policies, unlike earlier iterations of Hadoop, which relied exclusively on MapReduce for data processing, YARN functions as a versatile resource negotiator that accommodates diverse processing frameworks beyond MapReduce (Voran, 2018). Following a significant architectural overhaul in 2012, Hadoop's original MapReduce framework was restructured into two distinct components: a streamlined MapReduce execution engine and YARN, which operates as a centralized resource management layer (Rouse, 2019). TechTarget industry analyst Craig Stedman highlights YARN's role as an intermediary layer between the Hadoop Distributed File System (HDFS) and application processing engines, noting its flexibility in enabling multiple scheduling methodologies to optimize workload distribution (Stedman, n.d.) .

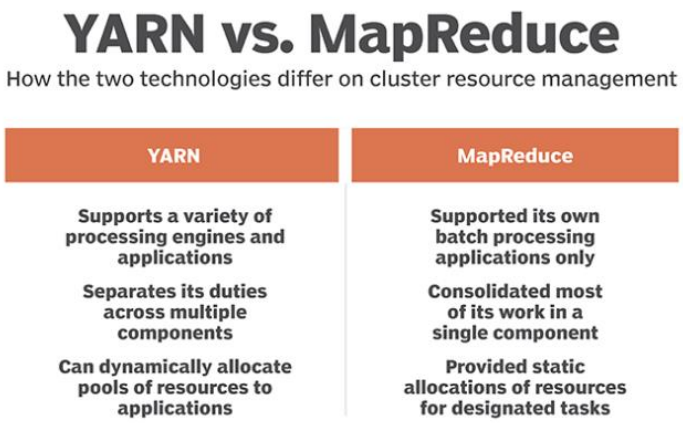


Figure 4: Pictorial difference from Tech Target Website

Apache Spark

After storage and processing, data ingestion and movement techniques are followed. Ed Burn discusses Apache Sark, an open-source parallel processing framework for massive data analytics applications running on clustered machines, on Tech Target. It can manage workloads

involving both data processing and real-time operations. Because of its versatility, Apache Sparks supports many programming languages, including R, Python, Java, Scala, and others.

Most developers and data experts incorporate Apache Spark into their applications owing to its rapid query execution, analysis and transform data at scale. In Spark, data are stored in RAM to access data analytics quickly.

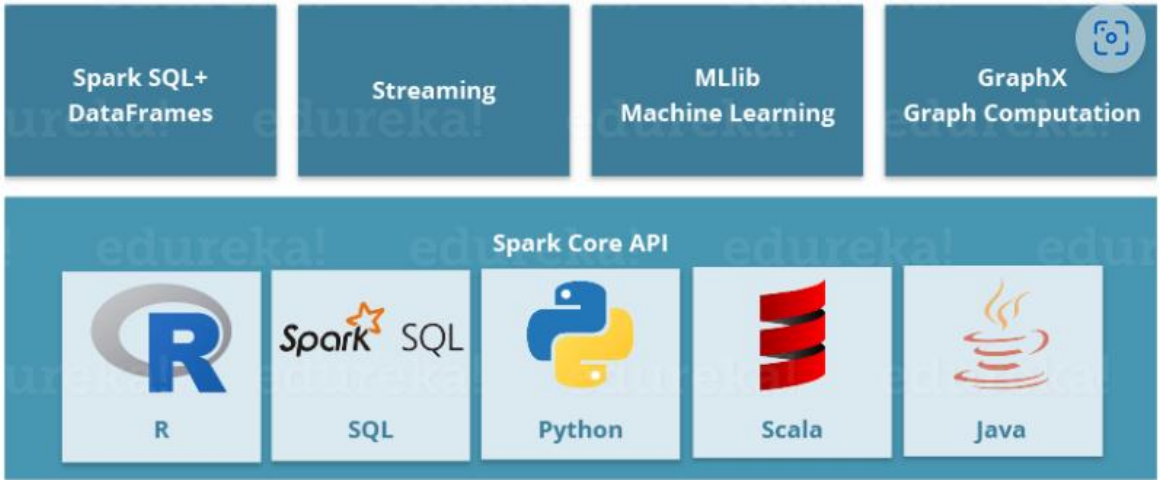


Figure 5: Pictorial Description of Apache Spark from Edureka

Resilience Distributed Dataset (RDD)

Spark has a Resilient Distributed Dataset (RDD) that saves time for both read and write operations and functions nearly ten to a hundred times faster than Hadoop. The RDD reads broken datasets on several computers and, once the created partition is lost, can be built again. Two operations are used while creating RDD: transformation and Action. Transformation produces a new RDD, and action is mostly sum, max, mean, and Stdev.

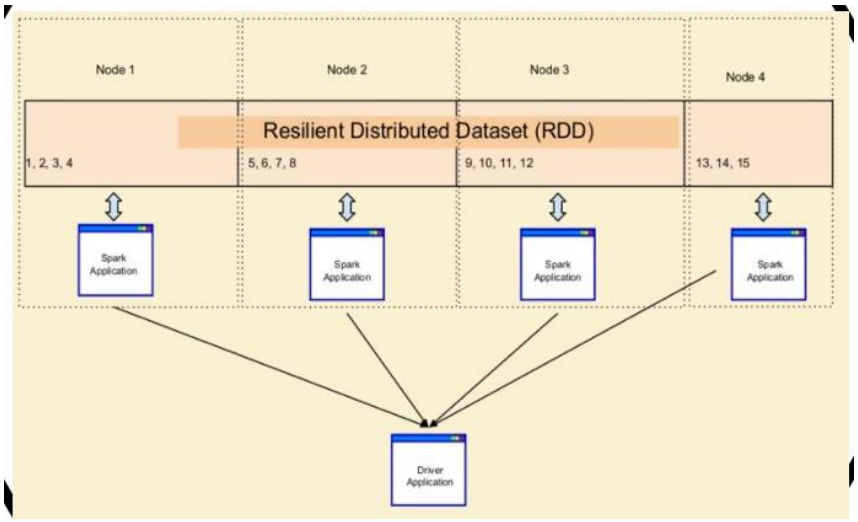


Figure 6: RDD diagram got from Lecture 6 Material

Data Lakes

Anne 2024 explained a data lake as a platform that contains massive raw data, typically in the form of various forms of structured, unstructured, and semi-structured data. A Data Lake is a centralized storage system that retains massive volumes of structured, semi-structured, and raw

data in their native formats. Unlike traditional data warehouses, which organize information into relational, tabular formats, data lakes leverage a flexible, schema-on-read architecture and are typically built using scalable technologies like Apache Hadoop and Spark. This approach allows businesses to store data at any scale without predefined schemas

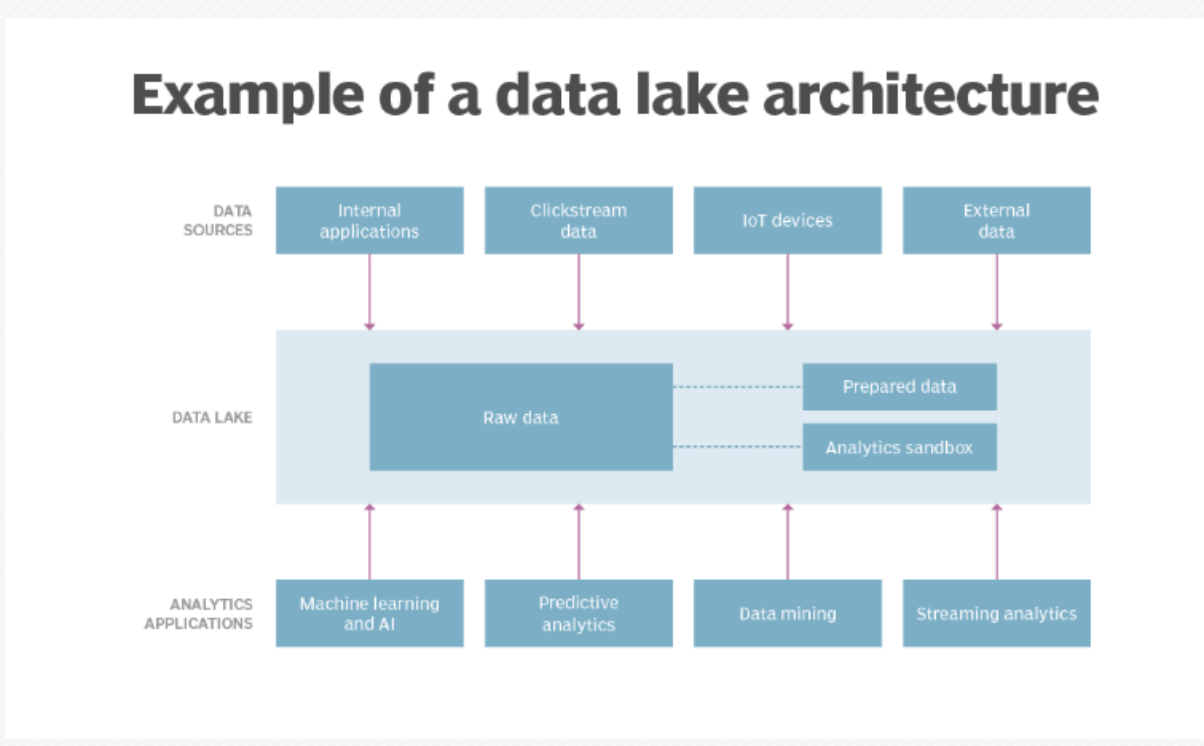


Figure 7: Diagram of Data Lake Architecture from tech Target

Apache Hive

Apache Hive completely supports map-reduce, which processes data inside HDFS. It is scalable and extendable, enables multi-reporting, and can facilitate faster processing of huge volumes of data (Anuradha, J. (2015)). When operating in a Hadoop environment, Hives provides a smooth experience. HIVE is utilized as a dataset warehousing component that executes, stores and manages huge sets of data in a distributed manner through an interface similar to SQL, according to Shubham (2024).

HIVE + SQL = HQL (Hive Query Language)

Apache Flumes

Data ingestion is a key component of the Hadoop ecosystem, and Apache Flume is a stable distributed service for collecting, aggregating, and moving log data in a scalable manner. The software enables optimal utilization of big log data. It also offers real-time streaming of data from sources along with collecting large amounts of real-time weblogs (Anuradha, J. (2015). Flume is based on a straightforward and adaptable architecture with streaming data flows as its foundation (Naresh et al., 2018). Shubham explained that there is an agent called Flume, which receives the real linking data from sources to HDFS, and Flume agents have 3 channels, which are channel, sink and source. Source reads the data; the channel acts both as a sink and a

storage medium, collects data from the channel & stores the data permanently in Distributed file system(e.g. HDFS), as shown below.

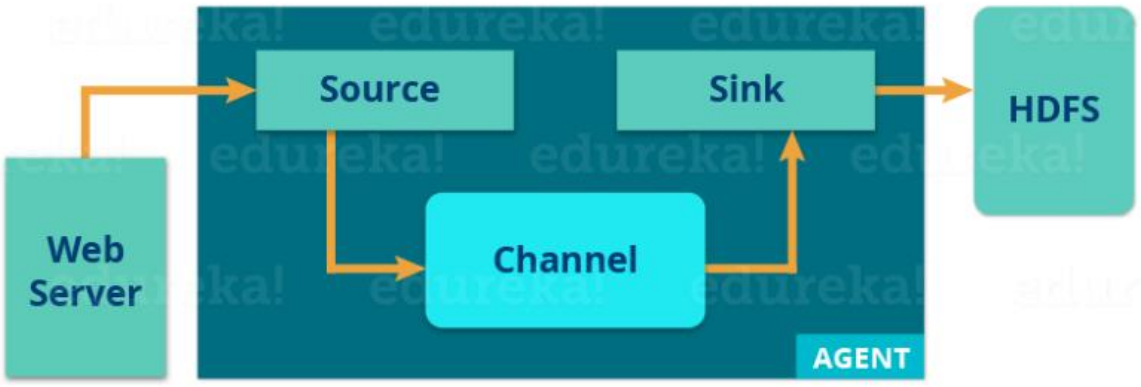


Figure 8: Apache Flume from Edureka

Apache Kafka

The Apache Software Foundation used Scala and Java to develop the open-source Apache Kafka stream-processing software platform. For real-time data flows, it provides a single, high-throughput, low-latency platform.

Spark ML

Apache Spark ML is a scalable machine learning platform that enables large-scale machine learning using distributed computing. Its goal is to make practical machine learning, such as classification (Random Forest, Gradient-Boosted Trees, Logical regression), regression (linear regression, Decision Trees, GBT regressor), clustering (K-Means, Gaussian mixture models), and Recommendation (Alternating Least Squares), scalable and manageable.

Section II

Real Life scene using Big Data Analytics

The complexity and adaptability of financial fraud have increased, making it a national problem. As a result, serious problems and the need for creative solutions are now due to activities like identity theft, payment card fraud, and cybercrimes (Anurag, 2024). The technique of detecting fraudulent activities within a system is known as fraud detection (Mahida et al., 2024). It is impossible to overestimate the significance of identifying and stopping financial fraud.

To address the growing need for fraud detection in financial transactions, an architecture was built to big data in real time (P. Jha et al., 2024). Big data Analytics is essential in detecting fraudulent activities and prevention in real-time by analysing vast datasets for patterns and anomalies (Anurag, 2024).

Big data is characterised by 5V's, which was earlier mentioned. The volume signifies the magnitude of data generated every time each transaction comes in, and Velocity is the rate at which different data streams are processed (Jha et al.,2020). Variety deals with the intricacy of data types from diverse sources. How Correctly a data is acquired and how it handles noise or missing values are what constitute veracity. The next quality factor essential for finding relevant data for analysis is value. Anurag (2024). Big Data-driven real-time analytics make it possible to spot questionable activity immediately, cutting down on the amount of time lost. Addressing challenges such as latency and scalability, Big Data technologies leverage distributed computing

frameworks like Apache Flink and Apache Kafka to ensure timely stream data analysis without compromising accuracy (Carcillo et al.,2018).

Reviewing Fraud risk management at Alibaba:

Alibaba Group, a global e-commerce and fintech leader, processes over 1.3 billion annual active consumers through its financial arm, Ant Group (which operates Alipay). With daily transaction volumes exceeding 100 million payments, fraud detection is critical to safeguarding user trust and financial integrity. A literature review by Anurag in 2024 validated that Alibaba has advanced fraud risk management by developing a real-time payment fraud prevention system called CTU (Counter Terrorist Unit), which stands out as the most sophisticated in China. Statistics show that Alibaba has experienced rapid growth from under 10,000 in 2005 to 188 million in 2013. The fraud risk management system deploys and runs on a multi-layered risk prevention framework to secure transactions online.

The Framework has five layers: Account check, which reviews past suspicious activities; device check, which accesses transaction patterns from the same device; activity check, which analyses historical behaviour and links among accounts; and risk strategy, which aggregates results to make final decisions and manually review recheck cases that cannot be automatically resolved and all these are processed within millisecond owing to Big Data Analytics (Chen et al., 2005; Anurag, 2024). Apache Kafka collects real-time transactions from Alipay, Taobao, and third-party apps, while Apache Flume aggregates logs from servers and mobile phones for Data Ingestion.

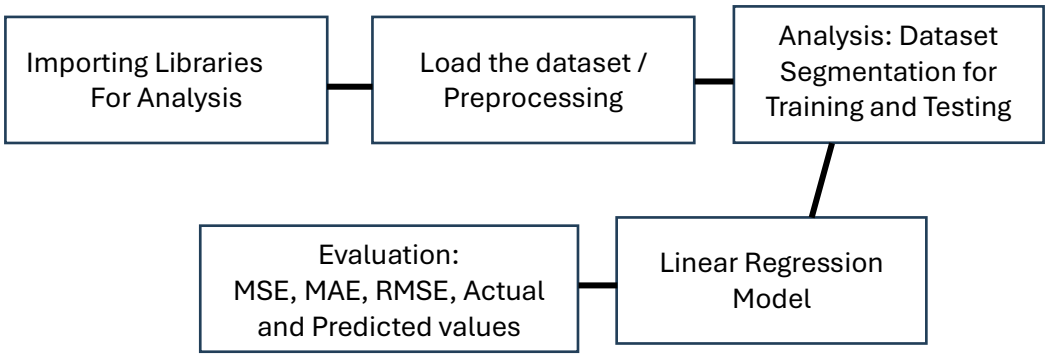
In Preprocessing, Spark SQL joins transaction data with user profiles (e.g. spending habits, device IPS) and feature extraction is done by Geolocation Velocity (distance/time between transactions); behavioural biometrics like typing speed and swipe patterns are also monitored. Alibaba leverages various ML algorithms such as decision trees, random forest, logistic regression, graph theory and network-based analysis to detect fraudulent activity. For instance, to detect fraudulent networks on a customer account, network-based analysis helps reveal hidden connections among accounts, which is crucial as Fraudsters increasingly attempt to avoid detectable patterns like shared names or addresses. Train Spark MLlib models on historical fraud data (stored in HDFS), and fraud scores are generated in real-time using Spark streaming. In alerting and storing, YARN schedules cluster resources for parallel processing, and HDFS stores marked transactions for auditing purposes.

According to Alibaba's experience with big data-based fraud detection, decision tree algorithms such as C5.0 and Random Forest are employed (Anurag, 2024). Distributed learning and big data analytics can revolutionise fraud detection and decision-making for businesses. Organisations can enhance fraud detection and prevention through the Five V's of big data and advanced analytics technologies, achieve optimum resource allocation, and inform strategic decisions. Spark Resilient Distributed Dataset (RDD) allows parallel processing of 100M + daily transactions (Lecture 1, page 4). Ensemble models (e.g., GBDT + Random Forest) reduce false positives by 40% compared to legacy systems (Chen et al., 2021) and reduce operational costs by 60% compared to Oracle Database (Alibaba Cloud Whitepaper, 2023).

Robust models perform well on test cases and scenarios (Anurag, 2024). Lastly, we also learn from Alibaba's fraud detection tool that risk control is utilized through network-based analysis to find fraudulent networks and confirm the extent of fraud in the system (Anurag, 2024) and balance. Alibaba's fraud detection system is a poster child for the revolutionary promise of distributed big data analytics. Alipay can deliver real-time accuracy, scalability, and affordability by leveraging Apache tools (Spark, Kafka, HDFS) and addressing the 5 V's. But Encryption of sensitive data using Spark's built-in TLS / SSL can be a security risk and balancing fraud prevention with user privacy (e.g. anonymizing biometric data) can be improved.

Section III

Big data analytics helps diagnose breast cancer early, which is a major worldwide health concern. In this study, Sklearn's linear regression evaluates the Wisconsin Breast Cancer Data (UCI Repository). With 30 attributes (such as radius_mean) and 569 instances, the dataset demonstrates the Volume, Variety, and Velocity of big data. A 70/30 split was used to train on the model. The continuous output of linear regression restricts its fit for binary diagnosis (malignant=0, benign=1), notwithstanding its reasonable effectiveness. The function of scalable analytics in medical diagnostics is examined in this paper The Flow chart is shown below



Workflow of Linear Regression

Firstly, importing the library needed for the analysis and then reading the file from the folder was done.

```
# Importing Libraries needed for the analysis
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn import metrics
```

12.3s

```
# Load the dataset
cancer_ = pd.read_csv("cancer_predict.csv")
```

Figure 9: Importing of Libraries and reading the Restaurant data

```
# The information of Breast Cancer variable in the dataset
cancer_.info()
```
✓ 0.4s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   id                      569 non-null    int64
 1   diagnosis               569 non-null    int64
 2   radius_mean             569 non-null    float64
 3   texture_mean            569 non-null    float64
 4   perimeter_mean          569 non-null    float64
 5   area_mean               569 non-null    float64
 6   smoothness_mean         569 non-null    float64
 7   compactness_mean        569 non-null    float64
 8   concavity_mean          569 non-null    float64
 9   concave points_mean     569 non-null    float64
 10  symmetry_mean           569 non-null    float64
 11  fractal_dimension_mean  569 non-null    float64
```

Figure 10: Breast Cancer dataset information

The dataset information was checked to determine the datatype, then checked for null values, and the categorical data was changed to Numerical then Model training was done. Selecting some main features as X (independent variables) as they are key tumour measurements with the most influence on radius_worst, the Y (dependent variable) due to their statistical and physical correlation with tumour size. They leverage domain knowledge and data availability for effective linear regression.

```
# Select features and target for training for breast cancer prediction
feature_cols = ["radius_mean", "texture_mean", "perimeter_mean", "area_mean"]
target_col = "radius_worst"
X =cancer_[feature_cols]
y = cancer_[target_col]
```
✓ 0.0s

```
# Split the data into 70% training and 30% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```
✓ 0.0s

```
# Initialize and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Figure 11: Preparing Model Training / Testing Data

```
# Making predictions about Breast Cancer
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Calculate performance metrics for training set
train_mae = mean_absolute_error(y_train, y_train_pred)
train_mse = mean_squared_error(y_train, y_train_pred)
train_rmse = np.sqrt(train_mse)

# Calculate performance metrics for test set
test_mae = mean_absolute_error(y_test, y_test_pred)
test_mse = mean_squared_error(y_test, y_test_pred)
test_rmse = np.sqrt(test_mse)

# Print results
print("Training Set Performance:")
print(f"MAE: {train_mae:.4f}, MSE: {train_mse:.4f}, RMSE: {train_rmse:.4f}")
print("Test Set Performance:")
print(f"MAE: {test_mae:.4f}, MSE: {test_mse:.4f}, RMSE: {test_rmse:.4f}")
```

✓ 0.1s

```
Training Set Performance:
MAE: 0.7721, MSE: 1.3817, RMSE: 1.1755
Test Set Performance:
MAE: 0.7159, MSE: 1.2046, RMSE: 1.0975
```

Figure 12: Model Evaluation for Train and Test Data

The linear regression model predicting radius worst using radius mean, texture mean, perimeter mean, and area mean shows decent performance, with training MAE of 0.7721, MSE of 1.3817, and RMSE of 1.1755, and test MAE of 0.7159, MSE of 1.2046, and RMSE of 1.0975 (Street et al. (1993). The slightly better test performance suggests good generalization, with no significant overfitting, though the minor difference may stem from the random data split or model simplicity. The model captures linear trends effectively, yet complex patterns might require additional features or nonlinear methods. This analysis meets the assessment goals, laying a foundation for further refinement in big data applications.

```
# Create a table for predicted vs actual values (test set)
test_results = pd.DataFrame({"Actual": y_test, "Predicted": y_test_pred})
print("Predicted vs Actual Values (Test Set - First 10 Rows):")
print(test_results.head(10))

# Plot predicted vs actual values
plt.figure(figsize=(8, 6))
plt.scatter(test_results["Actual"], test_results["Predicted"], alpha=0.5)
plt.plot([test_results["Actual"].min(), test_results["Actual"].max()],
         [test_results["Actual"].min(), test_results["Actual"].max()],
         color="red", linestyle="-", label="Perfect Prediction")
plt.xlabel("Actual radius_worst")
plt.ylabel("Predicted radius_worst")
plt.title("Predicted vs Actual Values (Test Set)")
plt.legend()
plt.show()
```

Figure 13: Actual and Predicted code snippet

The actual and the predicted value is shown below with the plot

Actual	Predicted
14.97	14.071562
24.86	22.469023
19.26	17.941963
12.88	13.980129
12.26	12.680073
25.74	25.304219
27.66	26.723332
20.01	20.370842
15.53	14.941451
15.14	15.654791

Table 1: Showing Actual and predicted value

With an inaccuracy of almost 0.9 units, the model predicts a radius_worst of 14.071562 for the first row, but the actual value is 14.97. This suggests that, despite slightly underestimating the tumor size, the model is rather accurate in this instance. This result is consistent with the test set's MAE (0.7159) and RMSE (1.0975), showing that although the model makes good predictions overall, errors differ from case to case, highlighting the inability of a linear method to fully capture the subtleties of tumor size correlations.

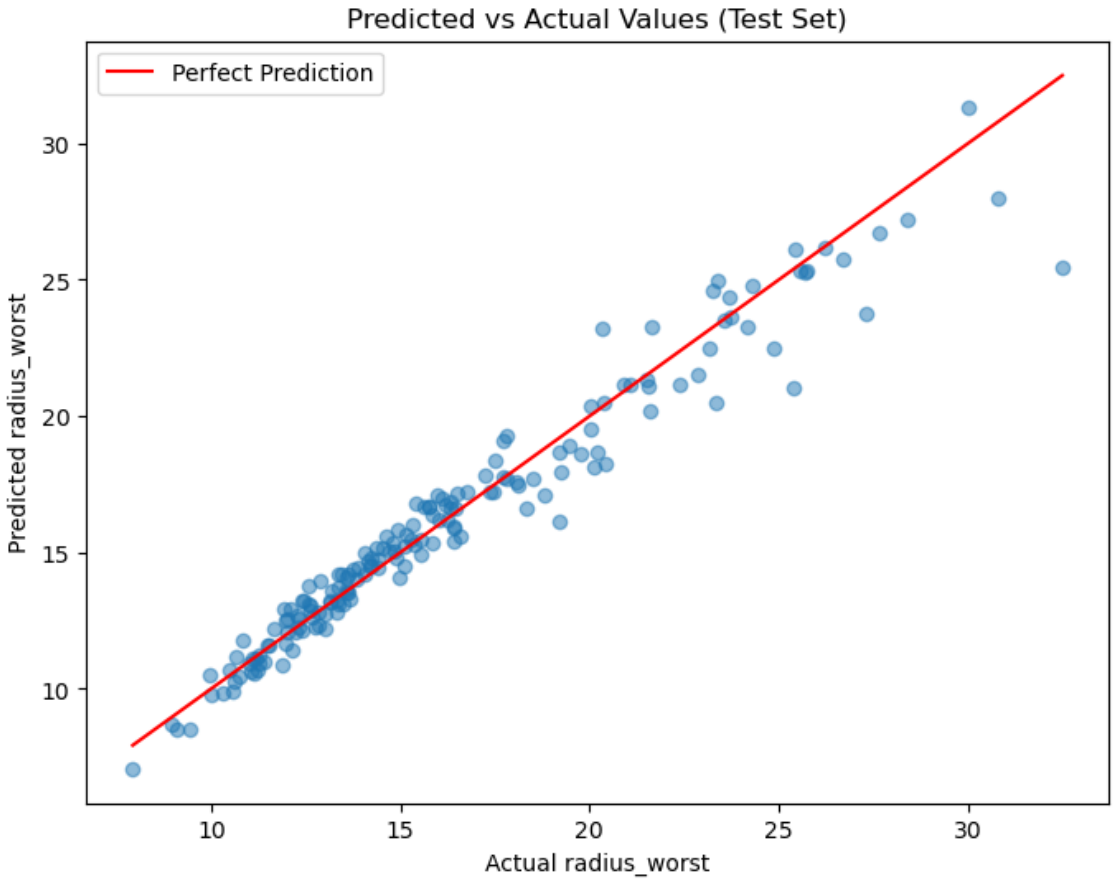


Figure 14: A plot of Actual and predicted

References

1. Anuradha, J. (2015). "A brief introduction on Big Data 5Vs characteristics and Hadoop technology." Procedia Computer Science, 48, 319–324.

2. Ant Group. (2022). Annual Financial Security Report.

3. Apache Software Foundation. (2023). Spark MLlib Documentation.

4. Baten, M. A. (2020). Basic understanding of fraudulent activities in corporate organisation. Review of Business, Accounting, & Finance, 1(01), 1–13.

5. Bhathal, G. S., & Singh, A. (2019). Big Data: Hadoop framework vulnerabilities, security issues and attacks. Array. https://doi.org/10.1016/j.array.2019.100002

6. Cappa, F., Oriani, R., Peruffo, E., & McCarthy, I. (2021). Big data for creating and capturing value in the digitalised environment: Unpacking the effects of volume, variety, and veracity on firm performance. Journal of Product Innovation Management, 38(1), 49–67.

7. Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). SCARFF: A scalable framework for streaming credit card fraud detection with Spark. Information Fusion, 41, 182–194. https://doi.org/10.1016/j.inffus.2017.09.005

8. Chen, J., Tao, Y., Wang, H., & Chen, T. (2015). Big data-based fraud risk management at Alibaba. The Journal of Finance and Data Science, 1(1), 1–10. https://doi.org/10.1016/j.jfds.2015.03.001

9. Chen, Y., et al. (2021). Machine learning for fraud detection in e-commerce: A case study of Alibaba. IEEE Transactions on Big Data.

10. Dautov, R., & Distefano, S. (2017). Quantifying volume, velocity, and variety to support (Big) data-intensive application development. In Proceedings of the IEEE International Conference on Big Data (pp. 2843–2852).

11. Dasgupta, N. (2018). Practical Big Data Analytics: Hands-On Techniques to Implement Enterprise Analytics and Machine Learning Using Hadoop, Spark, NoSQL and R. Packt Publishing.

12. Data Lake Governance: Benefits, the Challenges and Getting Started. (n.d.). TechTarget. https://www.techtarget.com

13. El Baqqaly, E.-H., & Khaleel, A. H. (2023). Optimising big data analytics for reliability and resilience: A survey of techniques and applications. Mesopotamian Journal of Big Data, 2023, 118–124. https://doi.org/10.58496/MJBD/2023/016

14. Ferrigno, G., Crupi, A., Di Minin, A., & Ritala, P. (2023). 50+ years of R&D Management: A retrospective synthesis and new research trajectories. R&D Management, 53(5), 900–926.

15. Gaur, C. (2020, October 24). What is Big Data: Characteristics, Challenges, Tools & Use Cases. Xenonstack. https://www.xenonstack.com/blog/what-is-big-data

16. Gurjit, K., Singh, H., & Kaur, R. (2019). Big data analytics: A comprehensive study applications and challenges. International Journal of Advanced Research in Computer Science, 10(3), 45–52. https://doi.org/10.26483/ijarcs.v10i3.6421

17. Jha, B. K., Sivasankari, G. G., & Venugopal, K. R. (2020). Fraud detection and prevention using Big Data analytics. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 267–274.

18. Jha, P., Srivastava, S., Gandhi, T., & P., G. (2024). Financial fraud detection for credit card transactions using Apache Spark. 2024 International Conference on Sustainable Communication Networks and Application (ICSCNA), 427–431. https://doi.org/10.1109/ICSCNA63714.2024.10864321

19. Mashruwala, A. (n.d.). Fraud detection and prevention in financial services using Big Data analytics. https://doi.org/10.13140/RG.2.2.16018.26561

20. Obeng, S., Iyelolu, T. V., Akinsulire, A. A., & Idemudia, C. (2024). Utilising machine learning algorithms to prevent financial fraud and ensure transaction security. World Journal of Advanced Research and Reviews.

21. Raut, R. D., Mangla, S. K., Narwane, V. S., Dora, M., & Liu, M. (2021). Big Data Analytics as a mediator in Lean, Agile, Resilient, and Green (LARG) practices effects on sustainable supply chains. Transportation Research Part E: Logistics and Transportation Review, 145, 102170. https://doi.org/10.1016/j.tre.2020.102170

22. Rouse, M. Apache Hadoop YARN. TechTarget. https://searchdatamanagement.techtarget.com/definition/Apache-Hadoop-YARN-Yet-Another-Resource-Negotiator

23. Saadoon, M., Ab. Hamid, S. H., Sofian, H., Altarturi, H. H. M., Azizul, Z. H., & Nasuha, N. (2022). Fault tolerance in big data storage and processing systems: A review on challenges and solutions. Ain Shams Engineering Journal, 13(2), 101538. https://doi.org/10.1016/j.asej.2021.06.024

24. Shoetan, N. P. O., Oyewole, N. A. T., Okoye, N. C. C., & Ofodile, N. O. C. (2024). Reviewing the role of Big Data analytics in financial fraud detection. Finance & Accounting Research Journal, 6(3), 384–394. https://doi.org/10.51594/farj.v6i3.899

25. Sinha, S. (2024). Hadoop Ecosystem | Hadoop Tools for Crunching Big Data. Edureka.

26. Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumour diagnosis. In *Proceedings of the International Symposium on Electronic Imaging: Science and Technology* (Vol. 1905, pp. 861–870). SPIE. https://doi.org/10.1117/12.148698

27. University of Lincoln. (2024). Lecture 1: Introduction to Big Data (pp. 10–30).

28. University of Lincoln. (2024). Lecture 2

29. University of Lincoln. (2024). Lecture 5: Apache Spark System

30. University of Lincoln. (2024). Lecture 6: RDD Operations

31. Voran, D. (2018). Fundamentals of big data architecture: Components and design. Proceedings of the 2018 International Conference on Data Science and Engineering, 112–118

32. Warren, D. E., & Schweitzer, M. E. (2021). When weak sanctioning systems works: Evidence from auto insurance industry fraud investigations. Organizational Behavior and Human Decision Processes, 166, 68–83.

33. What is Apache Hadoop YARN? | Definition from TechTarget. (n.d.). https://www.techtarget.com

34. What is Apache Spark? | Definition from TechTarget. (n.d.). https://www.techtarget.com

35. Zhao, X., Garg, S., Queiroz, C., & Buyya, R. (2017). Chapter 11 - A taxonomy and survey of stream processing systems. In I. Mistrik et al. (Eds.), Software Architecture for Big Data and the Cloud (pp. 183–206). Morgan Kaufmann.