Journal of Theoretical and Applied Information Technology
31st July 2024. Vol.102. No. 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PERFORMANCE COMPARISION OF APACHE SPARK AND SEQUENTIAL PROCESSING ON MACHINE LEARNING CLASSIFICATION ALGORITHM

N. SUDHAKAR YADAV[1], G. SURESH REDDY[2,*], A. SREENIVASA RAO[3], GANDLA SAI DHEERAJ RAO[4], MUKESH SAI PRODUTUR[5], P. NAYAB RASOOL KHAN[6], VALLAMKONDA ADARSH[7]

[1] Associate Professor, Department of Information Technology, CBIT Hyderabad, Telangana, India

[2]Professor, Department of Information Technology, VNRVJIET Hyderabad, Telangana, India

[3]Assistant Professor, Department of Information Technology, VNRVJIET Hyderabad, Telangana, India

[4,5,6,7] Department of Information Technology, VNRVJIET Hyderabad, Telangana, India

E-mail: gali.sureshreddy@gmail.com

ABSTRACT

Big data is a large collection of useful but often unstructured data. Machine learning uses this data to understand patterns and create models for analytical applications. Processing big data can be time-consuming, which is where frameworks like Apache Spark come in to help. These processing tools make real-time analytical applications more efficient and accurate. For example, credit card fraud detection uses big data frameworks to analyze transactions and predict whether they are fraudulent or valid based on certain attributes. This paper focuses on using Apache Spark for credit card fraud detection and compares its performance with sequential processing. The dataset used contains various features and over five lakh records labeled as fraud or valid transactions, stored in HDFS. The dataset is processed using the classification algorithm logistic regression in Spark's in-memory allotment, while the same dataset is processed sequentially and stored on the local system for comparison purposes. Performance comparisons are made based on metrics like RAM, CPU, network, disk usage monitored using Prometheus and Grafana monitoring tools. As the dataset size increases, Spark is expected to perform more efficiently compared to sequential processing. The user-defined implementation of logistic regression involves varying the threshold parameter value for equal sensitivity and specificity compared to the general threshold value which results in positive increases in accuracy, precision, sensitivity, specificity, and f1-score.

Keywords: *Big Data, Sequential Processing, Spark, Machine Learning, Logistic Regression*

1. INTRODUCTION

From the beginning of this century, there has been a rapid advancement in technology. One of the core foundations for this advancement is data. Data can be anything like height, weight, marks of students, medical history of patients, etc. In this ongoing change in technology, various applications have been built, these applications release a lot of digital traces which are abundant but are quite useful. This can be used in business analysis, recommendation systems, etc. Data is required to make applications more human-centric.

Data is very important and its storage and analysis are essential. Relational databases can store small-sized and structured data and process it but data is increasing at a very rapid rate and is generally unstructured, present in the form of images, audio etc. Here, big data will play as the solution. Big Data is any data that is too big to be stored in personal systems.

Hadoop is a famous big data application. Hadoop has an ecosystem that provides all the tools to perform all the tasks. For storage of data, HDFS is used. HDFS stands for Hadoop Distributed File System which is a file system used for storing data in a distributed manner [1]. For processing MapReduce is used, it consists of mapper and reducer functions that will perform the necessary computations, these computations are performed in the form of key and value pairs [2, 3, 4].

However, the performance of MapReduce is generally said to be time-consuming because of its

disk storage hence this led to the beginning of Apache Spark. It has a master-slave architecture and can perform various action and transform methods. Its faster nature is associated with the in-memory computations that it does. It is said that spark is about 100 times faster than MapReduce.

Machine Learning is a study under Artificial Intelligence that deals with processing data and analyzing patterns in them to form a model in order to perform predictions, forecasting and clustering. There are many supervised, unsupervised and semi supervised algorithms which can be used for a variety of real-time applications. The performance of the model is seen based on metrics like accuracy, precision, sensitivity, specificity, and f1 score which are calculated based on true positive, true negative, false positive, and false negative values [5].

The main focus this work is comparison of machine learning algorithms in spark and sequential environments. The limitation of this work is, not applied on improved machine learning algorithms. In future work, will use and compare the improved machine learning algorithms,

The empirical analysis is deconstructed in Section 4, which is followed by a discussion in Section 5. Section 6 contains the conclusion.

2. RELATED WORKS

R. Swathi et al. [6] talks about the data being generated in the real time events of various social networks like Twitter, Facebook. To visualize the data, Graph processing algorithms like Page rank are used. The Hadoop framework incorporates HDFS and MapReduce whereas in spark RDD and DAG are utilized. Spark has ML libraries called MLlib for processing the data using machine learning. In this paper Logistic Regression Performance has been compared among two frameworks Hadoop and Spark, Conclusions were drawn that Spark has outperformed because of the data storage is done in the memory and thus takes less time and less iterations to process the data than the MapReduce because of the in-memory processing. Yassine Benlachmi et al. [7] talks about performance comparison between Hadoop and Spark on word count algorithms. In this the authors have taken four files of different sizes (202MB, 137MB, 72MB, 34MB). They have applied a map function first followed by a reduce function to calculate the word count in each file. Map function takes more time than reduce function. They used the Scala programming language in spark which decreases the number of lines of code for calculating word count. Spark performs 100 times faster than Hadoop in memory operations. The time taken by spark to process the data is 10 times faster than Hadoop. In conclusion Hadoop performs better when dealing with the larger datasets whereas spark is a better alternative when it comes to scalability and speed for real time streaming applications.

P. Natesan et al. [8] describes a MR-MLR model meaning Multivariate Linear Regression which was implemented in MapReduce, to explore the feature of parallel programming. 4 various-sized dataset was taken which were based on power plant energy data, wave energy data, data which told about superconductivity and on topics like audio release in a set of years. The individual dataset was partitioned, during training, Mapper was used to calculate coefficients and intercepts and Reducer did an average of above, then created a model. During prediction splits were made, Mapper was used to predict and Reducer was used to average the performance metrics. MAE, RMSE, and R2 were used for analysis. Standalone and MR-MLR model had similar values of metrics. Performance was also observed for various splits of data. The influence of train-test split was seen where an increase in training data showed improving results.

F. Ouatik et al. [9] have worked on student orientation, which is a method to understand the past of the student along with his skills to find the right career path for the student. Generally taking place physically, it was digitized and Hadoop along with MapReduce was implemented. A dataset was made with rows containing the student data and columns containing the marks of the student in subjects like math, physics, languages etc. Hadoop is a big data tool, it has HDFS which was used for storing the dataset and for processing and analyzing the future career field, MapReduce was used. Model was made based on neural networks, kNN and naive bayes. Hadoop cluster was made with three computers as datanode and one computer as namenode. MapReduce was done with nineteen mappers and a reducer. Naive bayes had the highest accuracy and lowest computational time.

Md. Nowraj Farhan et al. [10] highlights the various Apache tools used to analyze Twitter data. Apache flume was used in taking data from Twitter to HDFS. Hadoop's MapReduce and Spark were used to analyze this data. Using these tools, the most tweeted programming language was found. Performance was seen on a singular node and on the cumulation of nodes or a cluster. On different-sized datasets, run-time was seen in MapReduce and Spark, which showed Spark performed much faster. As blocks decreased, run-time decreased for only MapReduce. As one by one slave nodes were removed, in

MapReduce time increased, and at one point computation couldn't be performed, similar was the result in Spark. Both perform better in clusters than single nodes.

Sujala D. Shetty et al. [11] has extensively worked in Spark and has used its libraries like spark streaming and MLlib. A machine learning algorithm called decision tree is used to make a model which was trained and tested using the processed. cleverland.data from data set based on problems or diseases caused in heart which belonged to UCI ML repos. This dataset had many features like max heart rate, blood pressure etc. along with label 0 indicating absence of heart disease and 1 for its presence. This data was kept in the amazon's cloud. Then spark streaming was used to connect to twitter, where the users used to send their information. After getting the results, a twitter direct message was used to send the result to the user. The model was made using the spark MLlib. Gini impurity was used in this model. Max tree depth along with maxBins was also found to get the best results.

Mohammed A. Rashid et al. [12] has worked on the performance analysis of Apache Hadoop and Apache Spark on large data sets using HiBench. They worked on 600GB of real time data generated on twitter, Facebook and other social media. The evaluation was done on the basis of the following metrics CPU bounds network bound and the disk bound. The processing claims that MapReduce outperformed well for smaller datasets of 1GB. And for data sizes of 40 GB or 100 GB and 200GB, Spark is faster than MapReduce. Different workloads such as Logistic Regression, Wordcount, Tera sort, Support Vector Machine, Matrix Factorization were considered to analyze the CPU utilization, memory, disk, and network input/output consumption at the time of job execution. And concluded that Spark was 2 times faster with word count and 14 times faster with the Tera sort workloads compared with MapReduce because of its in-memory processing Manal A Abdel-Fattah et al. [13] describes about utilizing spark and machine learning algorithms to predict CKD (chronic kidney disease). This uses a dataset that was gathered from UCI ML repositories. The dataset contains 400 samples with 24 features and a class label. The dataset is then subjected to feature preprocessing where null values and missing values are dealt with. Then as a next step the main features are extracted from the dataset using techniques such as RelieF and chi squared test which decreases the model's execution time and produces better results. Authors concluded that features obtained from the RelieF technique produces better results than chi squared test and full features. They have divided the dataset into training (80%) and testing (20%). The dataset is analyzed using the following ML algorithms like Logistic Regression, Decision Tree, Random Forest, SVM, Gradient Boosted Trees along with Naïve Bayes. These algorithms are implemented from the Spark MLlib library. The model was optimized by hyperparameter tuning using grid search with stratified K fold and kfold cross validation. Finally, they evaluated the model based on four metrics namely Accuracy, Precision, Recall and F1score. From the results obtained it was concluded that DT, GBT and SVM provides better performance with selected features than the other algorithms used in this paper.

Md Morouane Saim et al. [14] talks about cardiovascular diseases and the importance of early detection of this disease. Cardiovascular diseases are heart-based diseases. The aim was to find the category of risk of a cardiovascular disease for each patient in the next ten years. The dataset was taken from Kaggle and had features based on medical history and behavior of the people. Machine Learning algorithms like Logistic Regression, SVM and K-Nearest Neighbor were used to make predictions. Accuracy and f1 score were used to measure the effectiveness of the model and even the time for training was seen. SVM model presented the highest accuracy and f1 score and logistic regression had the least training time.

Pooja Tiwari et al. [15] talks about the credit cards and its related fraud and a model such as to detect if a credit card transaction is fraudulent or not. An introduction to credit card and its importance is given. There are various places credit card can be used. There is an increase in the usage of credit card which will further increase, with this increase of usage even the increase of credit card fraud is seen, various frauds are discussed like merchant fraud, abuse etc. Machine learning and deep learning algorithms were used to tackle this credit card fraud and create a model which tells if a transaction was fraudulent or not. Many datasets were taken from various sources and model were implemented on these. Further conclusions were drawn based on results, like SVM and kNN working better on small datasets, etc.

3. EXISTING SYSTEM

The traditional usage of databases is constrained to structured data. But in today's digital world most of the data generated is unstructured. Storing and processing big data is made possible using big data frameworks such as Hadoop and Spark. A larger

proposition of work in Big Data is done in MapReduce than in Spark. Data processing for real-time applications which requires usage of machine learning is done using inbuilt libraries such as Mahout in Hadoop and Mllib in Spark. These inbuilt libraries provide predefined machine learning algorithms which have limited usability. The Logistic Regression work is performed based on the threshold parameter, a threshold value equal to 0.5 is taken as default, which might not give the best accuracy. The top right of the graph used to gauge performance is where a model attempts to achieve high precision and high recall.

4. PROPOSED SYSTEM

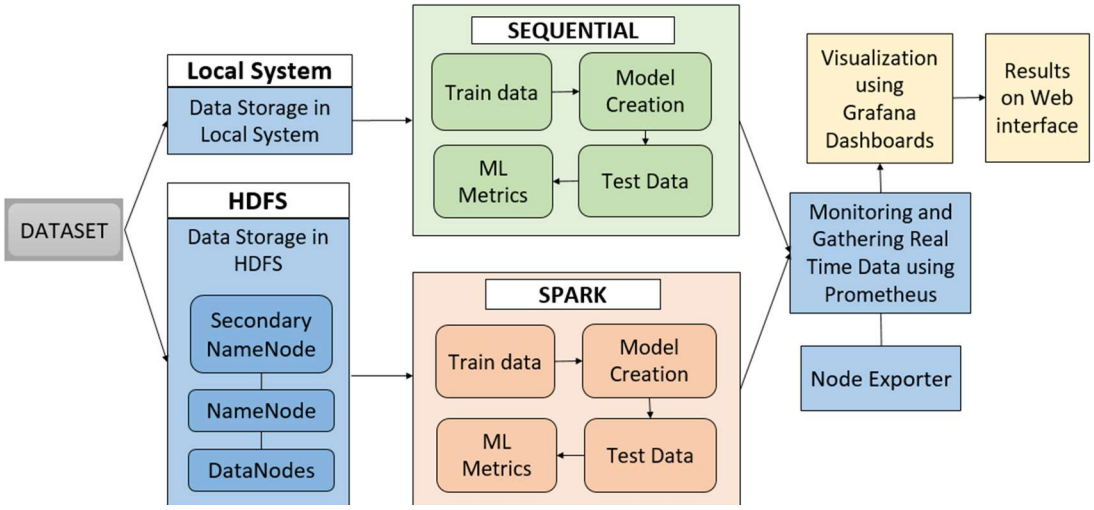


Fig. 4.1: Detailed Workflow Diagram Of The Proposed System

Based on the understanding of the existing system and based on research study, we aim at performing comparative analysis of Sequential processing and Apache Spark when a machine learning algorithm is executed. The proposed approach is a combination of user-defined implementation of logistic regression with Apache Spark, then comparing the performance when the same is processed sequentially. Credit card fraud data with various anonymous features and containing over five lakh entries is provided as input to the algorithm. The program being implemented is a user-defined Logistic Regression algorithm. This implementation will compare the default threshold with an alternative threshold which is based on that value with equal sensitivity and specificity. These thresholds will be compared based on metrics like accuracy, precision, sensitivity, specificity and f1score. The entire system performance during the execution of the algorithm is monitored by Prometheus and Grafana which will be running during the execution and will analyze the usage of network, RAM, CPU and disk. The data gathered by Prometheus and Grafana will be displayed on the React based Website.

4.1 Dataset used. The Credit Card Fraud 2023 dataset [16] belongs to the Digital Credit Card Fraud domain. This transaction is taken from European cardholders in the year 2023. It contains 5,68,630 transactions entered with 31 columns. The 31 features have a continuous unique identification, then V1, V2, V3, ..., V28 are independent anonymous features that could be location, time, etc. Then amount withdrawn is mentioned for each transaction and finally each record is assigned a label which is either 1 if fraudulent or 0 if valid [17, 18].

4.2 Algorithm used. Logistic Regression is a supervised machine learning classification algorithm. In this algorithm, the training data's independent variables form a linear relationship, which is given as an input to the sigmoid function, which releases a probabilistic value between 0 and 1. Here 0.5 threshold is used by default to classify, if the value is more than 0.5, its labeled 1 otherwise 0 [19, 20, 21]. Here the parameter threshold is altered to a value that produces equal sensitivity and specificity. To provide equal weightage to positive and negative values. Sensitivity tells how many positive values were correctly predicted as positive and Specificity tells how many negative values are correctly predicted as negative [22].

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

TP – True Positive TN – True Negative
FP – False Positive FN – False Negative

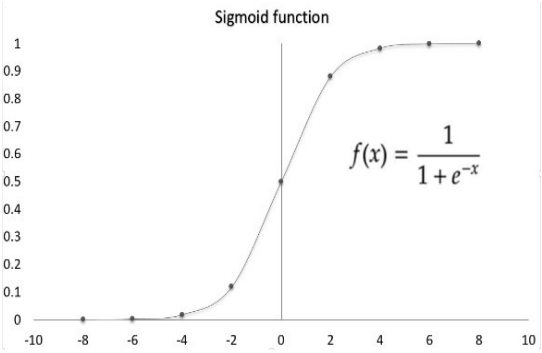


Fig. 4.2: Graphical Representation Of Sigmoid Function

4.3 Frameworks and Tools used. The Experimental setup includes execution and monitoring which are done on a Ubuntu system which is setup in Oracle Virtual Box. The initial setup before the execution includes installation of Anaconda Navigator, HDFS and Apache Spark. Node Exporter is added to Prometheus, which is connected to Grafana. It can be displayed on localhost. The execution and monitoring are performed in systems with different RAM: 8GB and 120 GB. Visual Studio code is used to display the data and graphs which are obtained at the time of execution.

Spark is a framework used for data processing. It is used for faster computational analysis of Big Data. It has modules that does SQL, ML and streaming [23, 24]. It has master slave architecture with a driver class and many executioner classes. Driver class is used to take input from the user, assigns tasks to executor class and keep track of tasks and executor class performs the tasks. It has the cluster manager which grants the resources to applications to complete the work. It does this by in-memory computations. Data reuse is done by creation of special data frames called Resilient Distributed Dataset (RDD) that is a data object that is cached in memory, and allows for being reusable in various operations of spark. It's a lazy evaluator as it does all the transformations and stores it in a special tree like structure and performs it only after seeing an action command. It has several action and transform methods [25, 26, 27]. It has PySpark to perform all python and big data related operations [28, 29].

Prometheus was originally built by SoundCloud, Prometheus is software that is used for monitoring and alternating toolkits. Prometheus stores timestamp for the metrics, its connected with Node Exporter. The purpose of this software is to monitor and store the system metrics which will be exported to Node Exporter. The data obtained can be analyzed and visualized effectively with Grafana [30].

Grafana made by the Grafana Labs, this software will graphically visualize the system metrics which is gathered by Prometheus. It is effective in monitoring live changes in the system metrics. Grafana provides a dashboard where all the data is displayed when Prometheus is connected using node exporter. Grafana supports querying Prometheus in a flexible way for users to analyze the data in detail [31].

Reactjs is a JavaScript library which is used in building user interfaces. Uses JSX. It is a collection of components which simplifies the process of creating interfaces. A web app created using Reactjs is used to display the comparative analysis of Spark and Sequential processing [32, 33].

5. IMPLEMENTATION

In systems of different architectures, Ubuntu is launched. Prometheus and Grafana begin the monitoring process once the system is launched. Firstly, in a sequential environment, logistic regression is implemented that includes, reading data from the local system, splitting it into train and test, train data is given to linear regression, test data is given to this model and the result is given to sigmoid function.
A range of threshold values are taken starting from 0.10 to 0.90, with a multiple of 0.10. For each threshold sensitivity and specificity are calculated and that threshold is chosen with equal sensitivity and specificity. This is plotted. Finally, ML metrics for 0.5 and altered threshold are calculated and shown in web interface.

HDFS and Spark are started. Credit card fraud data is stored in HDFS. In Spark, initially a spark session is created. The credit card fraud dataset is imported to the spark session. The data is the split for training and testing purpose. A linear regression model is obtained using the train data. Predictions for the test data are obtained by using the model. A sigmoid

function is then defined to find the sigmoid values. The sigmoid values are compared to a range of values which act as threshold. For each value, if the sigmoid is more than the threshold its labelled as 1 otherwise 0. Altered threshold is found and its ML metrics are compared with default 0.5 threshold. Graphs from Grafana and its observations are obtained and then displayed on the website.

6. RESULT ANALYSIS

The algorithms and performance was analyzed and tabulated below-0.5 was the normal threshold. In the range of thresholds from 0.10, 0.20, 0.30, … ,0.90, the better threshold was found to be 0.6. Threshold wise comparison of various frameworks is displayed in the tables below:

Table 6.1: Performance Metrics For 0.5 Threshold

0.5 Normal Threshold		
ML Metrics	Sequential	Spark
Accuracy	57%	57%
Precision	54%	54%
Sensitivity	99%	99%
Specificity	14%	14%
F1-score	70%	70%

Table 6.2: Performance Metrics For 0.6 Threshold

0.6 Better Threshold		
ML Metrics	Sequential	Spark
Accuracy	95%	95%
Precision	96%	96%
Sensitivity	93%	93%
Specificity	96%	96%
F1-score	95%	95%

It can be observed that the framework used for execution does not affect the performance of the algorithm. Both spark and sequential execution yield the same results.

Fig. 6.1: Bar Graph Of ML Metrics Of 0.5 And 0.6 Thresholds

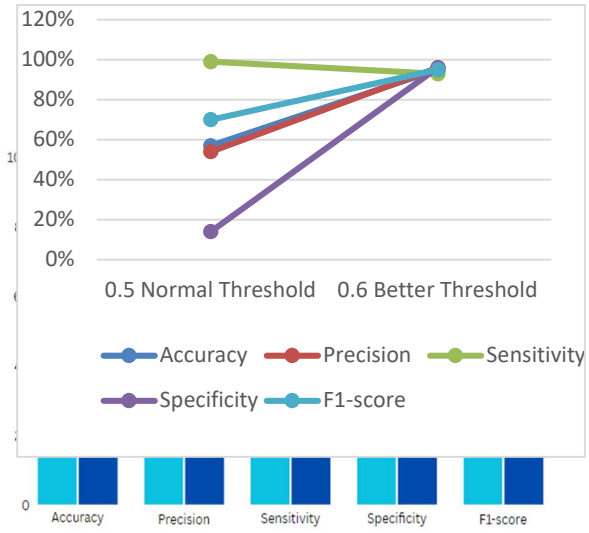


Fig. 6.2: Line Graph Of ML Metrics Of 0.5 And 0.6 Thresholds

The difference in frameworks appears in the system performance. The results obtained from the Prometheus and Grafana on an **8Gb RAM** system is shown

Table 6.3: System Metrics Obtained On 8Gb Ram System

System Metrics	Idle	Sequential	Spark
CPU Usage	~3%	~39%	~86%
Memory Usage	24%	45%	86%
Network Usage	62 Kbps	292 Kbps	816 Mbps
Disk Usage	28%	28%	28%
Time	-	~2.5 min	~3 min

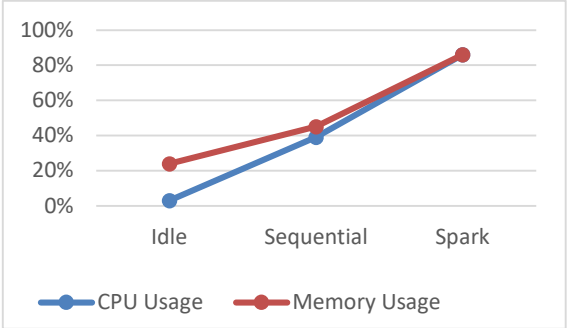


Fig. 6.3: CPU And Memory Usage Line Graph (8Gb System)

Journal of Theoretical and Applied Information Technology
31st July 2024. Vol.102. No. 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The graph demonstrates that the spark framework uses huge amount of resources for processing data. But it is only efficient when the dataset is very large. For datasets which can be easily processed by sequential system, spark would not be as efficient as it is expected to be.

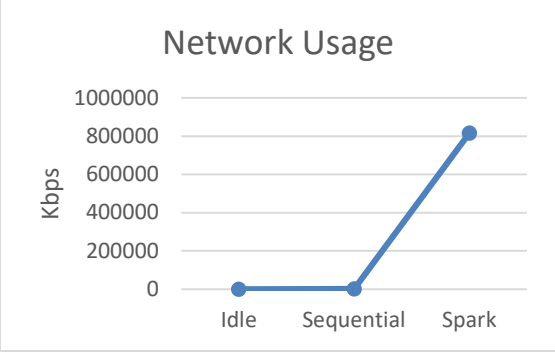


Fig. 6.4: Network Usage Line Graph (8gb System)

Since spark framework uses data from HDFS in this case, it performs huge amount of network operation for retrieving data from HDFS.

System Metrics obtained from an 8Gb RAM system.

The graphical comparison of overall system metrics during sequential and spark execution are shown below:

Fig. 6.5: System Metrics Of Idle System (8Gb System)

Fig. 6.6 Sequential Execution System Metrics (8gb System)

Fig. 6.7 Spark Execution System Metrics(8gb System)

Graphical representation of CPU usage:

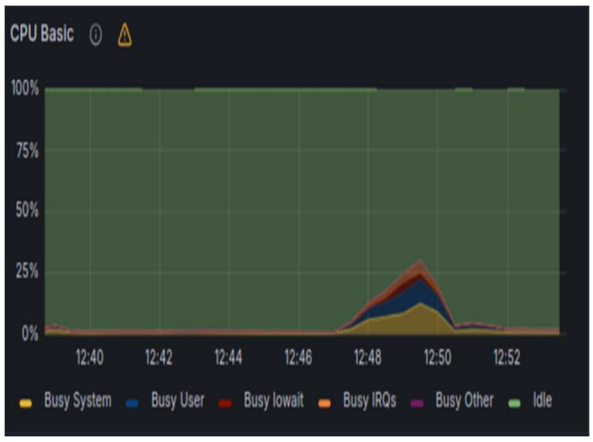


Fig.6.8: CPU usage in Sequential execution (8Gb system)

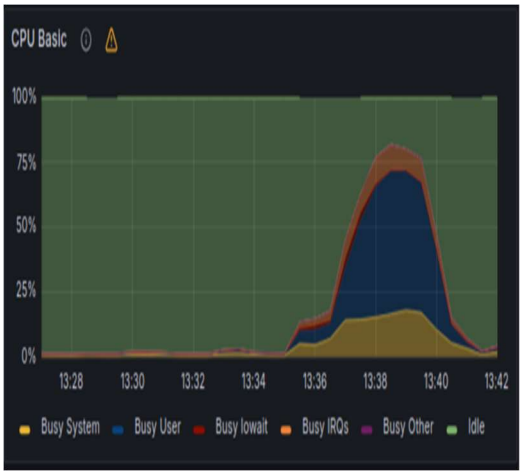


Fig. 6.9: CPU Usage In Spark Execution (8Gb System)

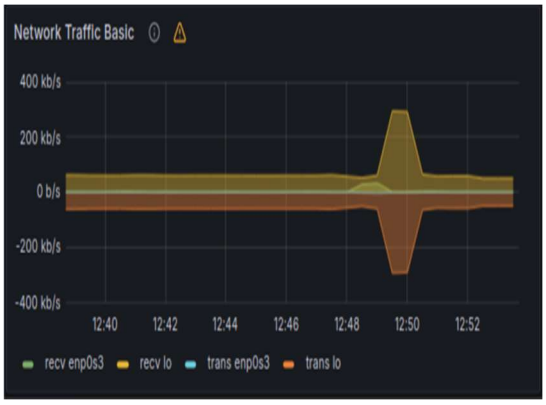


Fig.6.10: Memory Usage In Sequential Execution (8Gb System)

Graphical representation of Memory usage:

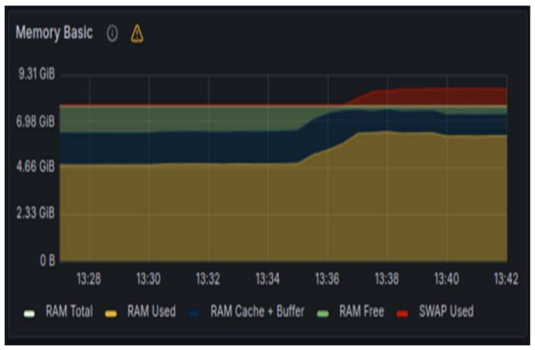
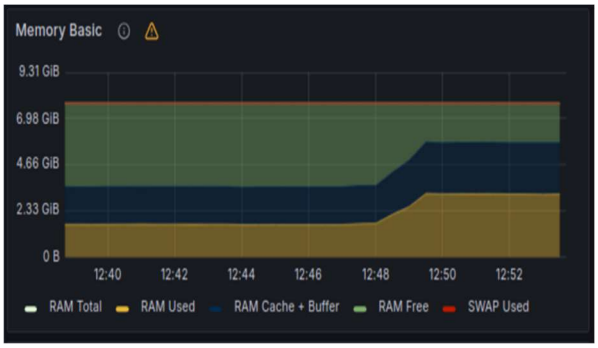


Fig.6.11: Memory Usage In Spark Execution (8Gb

As shown in the above graphs, spark uses huge amount of memory as it revolves around the concept of in-memory allocation. Sequential execution uses less memory when compared to Spark execution. Spark execution also uses SWAP memory that acts as virtual memory when system runs out of available physical memory.

Graphical representation of Network usage:

Fig.6.12:Network Usage During Sequential Execution



(8Gb System)

Fig.6.13:Network Usage During Spark Execution (8Gb System)

During sequential execution the dataset is present locally on the system so there is very less network usage. Where as during spark execution the dataset is imported from HDFS which results in high network usage.

The results obtained from the Prometheus and Grafana on an **120Gb RAM** system is shown below:

Table 6.4: System Metrics Obtained On 120Gb Ram System

System Metrics	Idle	Sequential	Spark
CPU Usage	~5%	~7%	~31%
Memory Usage	6%	7%	8%
Network Usage	61 Kbps	102 Kbps	982 Mbps
Disk Usage	79%	79%	79%
Time	-	~2 min	~3.5 min

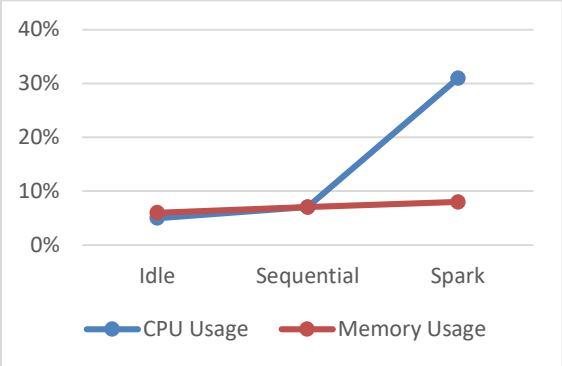


Fig.6.14: CPU And Memory Usage Line Graph (120 Gb System)

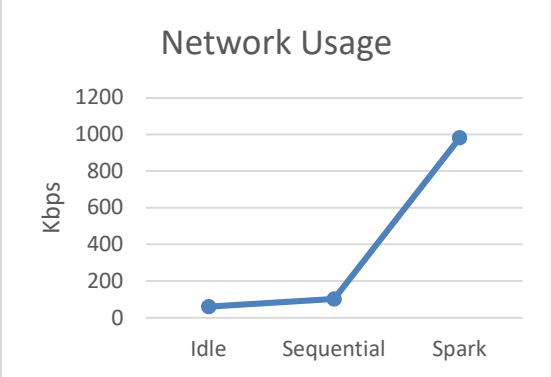


Fig.6.15: Network Usage Line Graph

6. FUTURE SCOPE

As a part of Big Data, there is no limit on the data that is taken as the input, so in the future, we would like to add more entries and explore other features. We would like to explore other techniques to improve the accuracy and also work on real time streaming data with the help of Apache Kafka.

With the above, we aim at getting very high accuracy. This paper focuses on credit card fraud, but in this digital world many other digital frauds are happening that are yet to be explored. Not only frauds but we other fields like medicine, economics could also be explored. While exploring other fields, Logistic Regression may not give the best results thus other algorithms in supervised and unsupervised domains can also be used depending on the application.

7. CONCLUSION

The combination of user-defined machine learning algorithm on big data framework has been achieved. Logistic regression was performed using the default threshold and then the results for better threshold were also obtained. This execution process was done in sequential and spark frameworks on systems with different capacities to analyze the performance of the system. The performance of the frameworks have been compared. The algorithm performance metrics were noted for both and there was an increase in these metrics for the model with altered threshold i.e. a high number of fraud transactions were classified as fraud and valid transactions were classified as valid.

These ml metrics were very similar in both Sequential and Spark but Spark in terms of performance used more resources than Sequential based on the analysis of Prometheus and Grafana. But the results were similar. This is because the dataset used contains

around 5 lakh records which is less than a real-time dataset. But as the size of dataset increases, the performance of Spark would also increase and yield efficient performance results. It is also be observed that when a system with more processing capacity is used, the difference in performance is less i.e spark uses more resources while giving same processing result as sequential processing.

REFERENCES

[1] Rahul Beakta, et al. "Big Data And Hadoop: A Review Paper". Baddi University of Emerging Sciences & Technology 2.2 (2015): 13-15, ISSN: 1694-2329.

[2] Shaikh Abdul Hannan, et al. "An overview on Big Data and Hadoop". Al-Baha University, International Journal of Computer Applications (0975-8887) 154.10 (2016): 29-35, DOI: https://doi.org/10.5120/ijca2016912241

[3] Mohd Rehan Ghazi , Durgaprasad Gangodkar, et al. "Hadoop, MapReduce and HDFS: A Developers Perspective". International Conference on Intelligent Computing, Communication & Convergence 48 (2015): 45-50

[4] V. Sajwan, V. Yadav et al. "The Hadoop Distributed File System: Architecture and Internals". International Journal of Combined Research & Development (IJCRD) 4.3 (2015): 541-544, ISSN: 2321-2241.

[5] Lidong Wang, Cheryl A Alexander et al. "Machine Learning in Big Data". International Journal of Mathematical, Engineering and Management Sciences" 1.2 (2016): 52-61, ISSN: 2455-7749.

[6] R. Swathi, Dr. R. Seshadri, "Performance Comparison of Machine Learning Algorithms in

Hadoop and Spark". IADS International Conference on Computing, Communications & Data Engineering (CCODE), SSRN, 2018, DOI: http://dx.doi.org/10.2139/ssrn.3167812

[7] Yassine Benlachimi, Abdelaziz El Yazidi, et al. "A Comparative Analysis of Hadoop and Spark frameworks using Word Count Algorithm". International Journal of Advanced Computer Science and Applications (IJACSA) 12.4 (2021): 778-788, DOI: 10.14569/IJACSA.2021.0120495

[8] P. Natesan, V. E. Sathishkumar, et al. "A Distributed Framework for Predictive Analytics using Big Data and MapReduce Parallel Programming". Mathematical Problems in Engineering (2023): 1-10, DOI: https://doi.org/10.1155/2023/6048891

[9] F. Ouatik, M. Erritali, et al. "Student Orientation using Machine Learning under MapReduce with Hadoop". Journal of Ubiquitous Systems & Pervasive Networks (IASKS) 13.1 (2020): 21-26, DOI: 10.5383/JUSPN.13.01.003

[10] Md. Nowraj Farhan, Md. Ahsan Habib and Md. Arshad Ali. "A study and performance comparison of MapReduce and Apache Spark on Twitter data on Hadoop cluster". International Journal of Information Technology and Computer Science (IJITCS) 10.7 (2018): 61-70, DOI: 10.5815/ijitcs.2018.07.07

[11] Sujala D. Shetty, Siddhanth D. Shetty, et al. "Applying Spark based Machine Learning model on streaming Big Data for Health Status Prediction". Computers and Electrical Engneering 65 (2018): 393-399, DOI: https://doi.org/10.1016/j.compeleceng.2017.03.009

[12] Mohammed A. Rashid, Ahmed1 , Andre L. C. Barczak , Teo Susnjak. "A comprehensive performance analysis of Apache Hadoop and Apache Spark for large scale data sets using HiBench." Journal of Big Data (2020), DOI: 10.21203/rs.3.rs-43526/v1

[13] Manal A Abdel-Fattah, Nermin Abdelhakim Othman, et al. "Predicting Chronic Kidney Disease Using Hybrid Machine Learning Based on Apache Spark". Computational Intelligence and Neuroscience 2022.15 (2022): 1-12, DOI: https://doi.org/10.1155/2022/9898831

[14] Md Marouane Saim, Hassan Ammor. "Comparative study of machine learning algorithms (SVM, Logistic Regression and KNN) to predict cardiovascular diseases". E3S Web of Conferences 351.21 (2022), DOI:10.1051/e3sconf/202235101037

[15] Pooja Tiwari, Simran Mehta et al. "Credit Card Fraud Detection using Machine Learning: A Study". arXiv 2108.1005v1 (2021), DOI: https://doi.org/10.48550/arXiv.2108.10005

[16] Dataset available: https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023

[17] Tej Paul Bhatla, Vikram Prabhu & Amit Dua, et al. "Understanding Credit Card Frauds". TCS (2003): 1-15

[18] Yashvi Jain, NamrataTiwari, et al. "A Comparative Analysis of Various Credit Card Fraud Detection Techniques". International Journal of Recent Technology and Engineering (IJRTE) 7.5S2 (2019): 402-407, ISSN: 2277-3878.

[19] J.S. Cramer, et al. "The Origins of Logistic Regression". Faculty of Economics and Econometrics, University of Amsterdam (2002), DOI: http://dx.doi.org/10.2139/ssrn.360300

[20] Alberto F. Cabrera, et al. "Logistic Regression Analysis in Higher Education: An Applied Perspective". School of Education, Handbook of Theory and Research 10 (1994): 225-256.

[21] A. S. Thanuja Nishadi, et al. "Predicting Heart Diseases In Logistic Regression Of Machine Learning Algorithms By Python Jupyterlab". International Journal of Advanced Research and Publications,University of Colombo 3.8 (2019): 69-74, ISSN: 2456-9992.

[22] Amelia Swift, et al. "What is sensitivity and specificity". Research Made Simple 23.1 (2020): 2-4, DOI: 10.1136/ebnurs-2019-103225

[23] Shaifali Yadav, et al. "Prediction Of Online Shopper's Buying Intention Using Algorithms Of Pyspark Mllib". International Journal of Current Science(IJCSPUB) 13.2 (2023): 61-70, ISSN:2250-1770, Available :https://rjpn.org/IJCSPUB/papers/IJCSP23B1363.pdf

[24] Michael Armbrust, Reynold S. Xin, et al. "Spark SQL: Relational Data Processing in Spark". MIT CSAIL (2015): 1383-1394, DOI: https://doi.org/10.1145/2723372.2742797

[25] Eman Shaikh, Iman Mohiuddin, Yasmeen Alufaisan, Irum Nahvi, et al. "Apache Spark: A Big Data Processing Engine". IEEE Middle East and North Africa COMMunications Conference (MENACOMM) (2019), DOI: 10.1109/MENACOMM46666.2019.8988541

[26] Amol Bansod, et al. "Efficient Big Data Analysis with Apache Spark in HDFS". International Journal of Engineering and Advanced Technology (IJEAT) 4.6 (2015): 313-316, ISSN: 2249-8958.

[27] Shweta Mittal, Om Prakash Sangawan, et al. "Implementing Machine Learning Algorithms On Spark". Guru Jambeshwar University of Science & Technology 11.5 (2021): 5267-5277, DOI: https://doi.org/10.28919/jmcs/5931, ISSN: 1927-5307.

[28] Rahul Pradhan,Praveen Kumar Mannepallli and Vikram Rajpoot, et al. "Analysing Uber Trips using PySpark". Materials Science and Engineering ICAMCM (2021): 1-9, DOI:10.1088/1757-899X/1119/1/012013

[29] HarshalS. Kudale, Mihir V. Phadnis, et al. "Data Analysis and Visualization of Olympics Using Pyspark and Dash-Plotly". International Research Journal of Modernization in Engineering Technology and Science 4.6 (2022): 998-1005, ISSN: 2582-5208.

[30] Arun Kumar,Vinutha B,VinayAditya B, et al. "Real Time Monotoring Of Servers With Prometheus and Grafana for high availability". International Research Journal of Engineering and Technology (IRJET) 6.4 (2019): 5093-5096, ISSN: 2395-0056.

[31] Abhishek Pratap Singh, et al. "A Data Visualization Tool-Grafana" Journal of Emerging Technologies and Innovative Research (2023), DOI: 10.1109/MENACOMM46666.2019.8988541

[32] Priya Maurya, Pankaj Keshari, Dr. Alok Katiyar, et al. "Web Development Using ReactJS". Journal of Current Research in Engineering and Science 6.2 (2023): 1-9, ISSN: 2581-611X.