International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

Efficient Big Data Machine Learning with Apache Spark: Comparative Evaluation of Regression and Classification Models

A Ganesh[1], R R Shantha Spandana[2], E Pavithra[3]

[1]P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur,
E-mail: alpakamganesh3@gmail.com

[2]Assistant Professor, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur,
E-mail: shanthaspandana@gmail.com, ORC-ID: https://orcid.org/0009-0003-4236-1250

[3] Assistant Professor, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur,
E-mail: epavithra526@gmail.com, ORC-ID: https://orcid.org/0009-0006-8871-4551

Abstract: The scale of big data requires machine learning (ML) systems capable of operating large amounts of data in little time. The implementation makes use of distributed computation and in-memory processing of Apache Spark to evaluate a variety of machine learning methods applied to regression and classification tasks. The experiments were conducted on three datasets, such as NYC Taxi Trip Duration, Netflix Prize, and Higgs boson, which were received through Kaggle. Linear Regression, Random Forest (RF), Gradient-Boosted Trees (GBT), Support Vector Regressor (SVR), and K-Nearest Neighbors (KNN) were compared by evaluating Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), training time and testing time. The classification methodology compared to classify data using Accuracy, Precision, Recall, F1-Score, AUC, and Confusion Matrix were LR, RF, GBT, SVM, and KNN. The findings demonstrated that both GBT and RF required a little more processing time and achieved higher accuracy and lower error rates. Selection of features using PCA improved the performance of the feature selection, and PCA-combined GBT reached 99 percentage classification accuracy and reduced training time. GBT achieved the lowest RMSE and MAE in regression tasks and was able to obtain the full prediction accuracy at the test data. Flask-based web interface was developed to make real-time predictions, which enable effective and accurate big data analytics.

"Index Terms: *Big Data, Biological system modeling, Data models, Cluster computing, Computational modeling, Sparks, Numerical models, Scalability, Predictive models, Public transportation".*

1. INTRODUCTION

Due to the accelerated growth of the digital era, the amount of data generated by social media, IoT gadgets, e-business sites, and in-house applications has grown exponentially. The current flood has ushered in the era of big data which consists of large volume, speed and diversity that have a huge limitation on the traditional ways of storing and processing data [7]. Companies are becoming more aware of the importance of using such datasets to gain a competitive edge, innovativeness, and make evidence-based decisions. With the world industry moving towards the data-focused operation, the challenge of extracting actionable insights by utilizing large and intricate information has become a serious issue [6].

ML, a core area or subfield of AI, is the ability of systems to discover patterns, build predictive models and improve performance without the explicit reprogramming. Within the framework of big data systems, ML allows predictive analytics, anomaly detection, and informed decision-making in domains like healthcare, finance, logistics, and customer engagement. Applications of machine learning on large data sets require distributed systems that offer concurrent computing capabilities, iterative learning, and optimal utilization.

The Apache Spark is a large data processing engine that is open source and meets these requirements through in-memory computing, fault tolerance and scalable cluster management [8]. Its architecture is significantly superior to traditional disk-based systems like Hadoop in the case of iterative workloads, which makes it particularly suitable in ML application [7]. The ability of Spark to speed up the training and inference of ML through distributed processing has been empirically tested to reduce latency and enhance throughput [1]. The studies of Spark scalability demonstrate that it is able to support the growing data

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

volumes without compromising the processing speed in the high-demand environment [2].

The optimization of resources is needed to improve the performance of Spark in an enterprise-level environment, with workload balance and cluster tuning being the key points of focus to realize the effective processing [3]. Studies that have been conducted on the implementation of Spark on private or hybrid clouds show promising results and prove that it can be deployed in any type of infrastructure without impairing its performance or reliability [4]. MLlib library of Spark supports large-scale predictive analytics pipelines, is resilient, precise, and flexible to real-time situations [5].

Enterprises can now process terabytes of mixed data, build smart models and scale analytics using Spark with ML. Such synergy will accelerate decision-making, enhance creativity, and data-driven strategies and make ML-enabled big data analytics an essential component of modern digital transformation [1][8].

2. LITERATURE REVIEW

There has emerged a growing body of literature examining the intersection of big data analytics, distributed computing, and ML, specifically the frameworks such as Apache Spark and ML library, MLlib. Ali, Abbod, Khaleel, Mohammed, and Sutikno [9] focused on the scalability of large-scale data analysis with the help of MLib and presented the empirical results of the Spark efficiency in working with large amounts of data and helping to make forecasts. Their study has pointed out how MLlib is also competent in handling iterative algorithms with better results than its traditional counterparts which has been attributed to the in-memory computing of Spark. The paper has highlighted the significance of proper utilization of resources, fault tolerance and distributed architecture in achieving low-latency analytics in a number of industrial applications. Also, Quinto and Quinto [10] provided a comprehensive overview of Spark and Spark MLlib with the basic concepts, APIs, and distributed ML operations. Their discussion covered advanced techniques, including gradient-boosted trees, distributed DL, and the aspect of integration with major libraries, which give practitioners a structured understanding of the use of Spark with regards to modern ML in large-scale environments.

The paper by Ip, Ang, Seng, Broster, and Pratley [11] has explored the use of big data and ML in agriculture especially crop protection. The authors studied the incorporation of large volumes of sensor, satellite imaging, and meteorological data into the distributed computing infrastructures like Spark to build predictive models of pest infestations and disease detection. The study proposed ML models that can be used to improve the decision-making process in precision agriculture with the focus on their ability to reduce losses in crops and maximize the use of chemicals by providing accurate and timely information. The work by Divya, Bhargavi, and Jyothi [12] reviewed the existing ML algorithms in the context of big data analytics and categorized them into supervised, unsupervised, and reinforcement learning methods, depending on their suitability by large datasets. Their study noted the issues such as data pretreatment, feature selection, and algorithmic efficiency and suggested distributed systems such as Spark MLlib to address computational bottlenecks and improve real-time processing in such industries as finance, healthcare, and e-commerce.

The study by Aziz, Zaidouni, and Bellafkih [13] is a comparative study of massive data processing based on the ML techniques, in this case, it is MLlib and Apache Mahout. Their study included real applications to evaluate performance, scalability and integration ease among platforms. They concluded that Spark MLlib has better support to the application of iterative algorithms and low execution time compared to Mahout, particularly where structured and semi-structured data are required. Their study showed the importance of resource allocation tools and in-memory processing in order to meet the high throughput needs of the big data systems at the enterprise level. Nair, Shetty and Shetty [14] proposed a ML model based on Spark application to predict health condition using streaming massive data. They developed real-time analytics pipelines, which are capable of receiving an uninterrupted stream of health-related data, gathered by wearable sensors, and then use classification algorithms to detect abnormalities and predict emergent critical situations. The study found out that Spark streaming with MLlib is an efficient tool to deal with dynamic data streams and guarantee a high predictive accuracy and low latency,

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

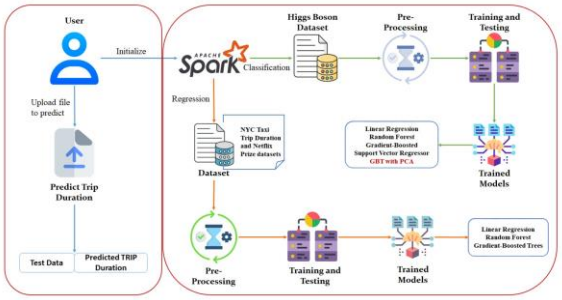
which makes it suitable in telemedicine and early warning systems.

Alsheikh, Niyato, Lin, Tan and Han [15] investigated the mobile big data analytics using DL [20] linked with Apache Spark. They have addressed the difficulties related to processing large volumes of data created by mobile networks and IoT devices, offering a hybrid architecture, which combined distributed processing of Spark with DL models to perform such tasks as user behavior prediction, network optimization, and personalized services. They highlighted the benefits of using the in-memory compute of Spark to further promote the training of DNN to reduce computational power and enable near real time analytics of mobile operators. Their approach demonstrated the ability of processing of unstructured and high-dimensional data streams by incorporating DL into distributed systems and thereby increasing the applicability of Spark to the next-generation mobile system.

Omar and Jumaa [16] provided a practical use of big data analysis using Apache Spark MLlib and Hadoop HDFS based on Scala and Java. Their study showed that Spark is being integrated with Hadoop ecosystem to build a robust system of data feed, storage, and ML. They developed case studies to illustrate how Spark MLlib can be used to train and predict scale models to address such concerns like fault tolerance and heterogeneous data management. Their results showed that Spark-Hadoop [18] provides the ability to store data and perform computations, which makes it a strong choice in analytics pipelines of enterprises of retail, energy, and governmental services.

3. MATERIALS AND METHODS

The proposed framework applies the capabilities of Apache Spark, namely, distributed computing and in-memory processing, to create and evaluate ML algorithms to handle the management of large datasets effectively. Three Kaggle datasets, including NYC Taxi Trip Duration, Netflix Prize, and Higgs boson, are used to address the problem of regression and classification. Regression algorithms (Linear Regression, RF, GBT, SVR and KNN) are evaluated based on the values of such metrics as RMSE, MAE, training time, and testing time. The classification is done with the use of LR, RF, GBT, SVM, and KNN, being checked with such measures as Accuracy, Precision, Recall, F1-Score, AUC, and Confusion Matrix. The feature selection technique is PCA that is used to improve input qualities and improve the performance of the algorithm. The system is implemented in Jupyter notebook with a web interface based on Flask that is used to predict and deploy efficiently.



"Fig.1 Proposed Architecture"

The architecture represented shows a workflow that uses Apache Spark to predict and categorize trip durations. The input data is entered by users, and it is prepared and then combined into the distributed framework of Spark. The regression of the NYC Taxi Trip and related datasets and classification of the Higgs boson dataset are handled by separate pipelines. The ML models (such as LR, RF, Gradient-Boosted Trees, and Support Vector Regression) are trained and tested according to dimensionality reduction conducted with the help of PCA in case of necessity. The expected outputs generated by trained models can enable scalable, efficient, and accurate transportation forecasting big-data analytics.

a) Dataset:

The dataset of NYC Taxi Trip Duration contains 729,322 records and 19 columns with information about a trip including pickup and dropoff points, time, vendor information, passenger numbers, and the duration of the trip and makes it possible to predict the time of the journey in New York City. Variables include continuous (coordinates, duration), discrete (vendor_id, passenger-count), datetime and categorical data. The Higgs boson dataset, which was received on Kaggle, provides particle physics measurements of signal and background classification, including a range of numerical features that are the readings of the detector, which is essential in training ML models in predicting and pattern recognition of high-energy physics events.

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

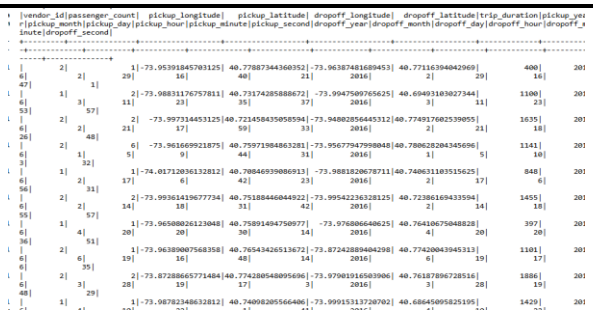


Fig.2 Data Collection

b) Pre-Processing:

Preprocessing stage is necessary to ensure the quality and reliability of the dataset. It involves cleaning up the data through filling in missing or inappropriate values and transforming features into appropriate models. Categorical variables are converted to numerical forms that can be applied in ML methods. In addition, the data is structured and normalised, which makes it appropriate to be used in the model. The stage ensures that the next learning algorithms receive high quality, standardized and clean data to ensure maximum effectiveness.

c) Training and Testing:

After processing, the data is divided into training and test data in order to determine the performance of the models efficiently. A significant portion (e.g. 80 percent) of the data is used as training to enable the model to identify patterns and relationships in the data. It splits the remaining part (20 percent) into testing to ensure that the model predicts the performance of the model impartially. This split helps in overfitting, performance evaluation on new data sets and helps in the optimization of the model parameters to ensure reliable and accurate prediction.

d) Algorithms:

Regression Algorithms:

Linear Regression (LR) – The duration of the NYC Taxi trips is a continuous variable that is predicted using Linear Regression to explain the association between the input data and the output using a best-fit line. It minimizes the sum of squared differences between the expected and observed values, which is a simple, understandable, and computationally efficient method of large datasets in the distributed Spark platform.

Random Forest (RF) – RF regression is a combination of many decision trees that are trained on random data and property sets in order to generate final predictions through averaging. This cancels overfitting and makes it more accurate, which makes it resistant to large high-dimensional data. The parallel processing of Spark ensures that RF handles the complex interaction of features.

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \quad (1)$$

Gradient-Boosted Trees (GBT) – Gradient Boosting Trees in regression The trees, which are built to solve the problem using gradient Boosting, are built one after the other and the tree at each stage corrects the errors of its predecessor. The technique is an enhancement method that improves accuracy and solves non-linear associations and minimizes forecasting errors. Within Spark, it can easily handle large volumes of data and achieve a high predictive performance with relatively small values of RMSE and MAE.

Classification Algorithms:

Logistic Regression (LR) – Logistic Regression is used to carry out binary classification whereby input data is converted into probabilities by a logistic regression and threshold is used to determine the classes. It is fast, interpretable, and highly suitable when it comes to large scale classification in Spark.

$$\hat{y}_i = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}} \quad (2)$$

Random Forest (RF) – RF builds as many decision trees as possible in the classification and classifies the object or data using majority voting. Its robustness, the ability to deal with high-dimensional properties and scalability in Spark make it the best one when it comes to doing accurate classification on large-scale datasets.

Gradient-Boosted Trees (GBT) – GBT as a classification method builds trees sequentially in a correctional manner to improve accuracy and clarify complex associations in the data. It can perform well with large scale datasets using the distributed architecture of Spark with good precision, recall and F1-Score.

Support Vector Machines (SVM) –SVM classify data according to the discrete classes by classifying the data with the help of identifying a suitable hyperplane and subsequently using kernel functions to perform non-linear separations. It is high dimensional and

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

scalable in Spark where large sets of classification data are required.

$$minimize\ \frac{1}{2}\ ||W||^2 + C\sum_{i=1}^{n}\xi_i \quad (3)$$

GBT with Principal Component Analysis (PCA) – PCA reduces the number of dimensions, which is accomplished by identifying the most relevant attributes before GBT training and therefore reduces the computation time and improves the accuracy of the model. Such a combination enhances the effectiveness of classification especially on large-dimensional data, within the scalable Spark environment.

4. EXPERIMENTAL RESULTS

RMSE: Root mean squared error (RMSE) is a measure of the mean error of a statistical model in predicting its values and comparing them with the values actually observed. It is the standard deviation of the residuals, mathematically. Residuals refer to the difference between the regression line and the data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}||y(i) - \hat{y}(i)||^2}{N}} \quad (4)$$

MAE: Absolute mistake The extent of error in measurements. The difference between the measured and the true value. As an example, assuming that your weight is 89 pounds, and a scale shows 90 pounds, the absolute error will be 90 lbs 89 lbs = 1lb.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (5)$$

Accuracy: Accuracy of a test is defined as the ability to differentiate patient and healthy cases correctly. To determine the test accuracy, a calculation of the proportion of the true positive and true negative cases assessed is to be carried out. This mathematically can be expressed as:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} (6)$$

Precision: Precision measures the percentage of correctly identified cases of those that are positive. The precision calculation formula can therefore be presented in the form as follows:

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{False Positive}} (7)$$

Recall: Recall is a ML metric that measures the ability of a given model to identify every relevant instance of a certain class. Proportion of the correctly predicted positive observations to the total actual positive is the proportion that provides the information on the effectiveness of a model in detecting instances of a certain class.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} (8)$$

F1-Score: F1 score is a value that is used to assess the precision of a ML model. It combines the recall and the precision of a model. Accuracy measure is used to measure the number of true predictions made by a model on the whole dataset.

$$F1\ Score = 2 * \frac{Recall\ \mathrm{X}\ Precision}{Recall + Precision} * 100 (9)$$

The performance analysis in Table 1 shows that Gradient Boosting has achieved the best performance and surpasses other algorithms in the RMSE and MAE performance, therefore, signaling a higher predictive accuracy at a minimum testing time.

The following table (Table 2) is a performance analysis using Accuracy, Precision, Recall and F-Score to compare the performance of models which GBT using PCA outperforms the others by achieving the highest values and higher classification overall.

Table.1 Performance Evaluation Table – Regression

Algorithm Name	Train Time	Test Time	RMSE	MAE
Logistic Regression	40.48	0.08	6.27	4.58
Random Forest	91.03	1.27	1.69	0.92
Gradient Boosting	**100.61**	**0.09**	**1.23**	**0.73**

Graph.2 Comparison Graph – Regression

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

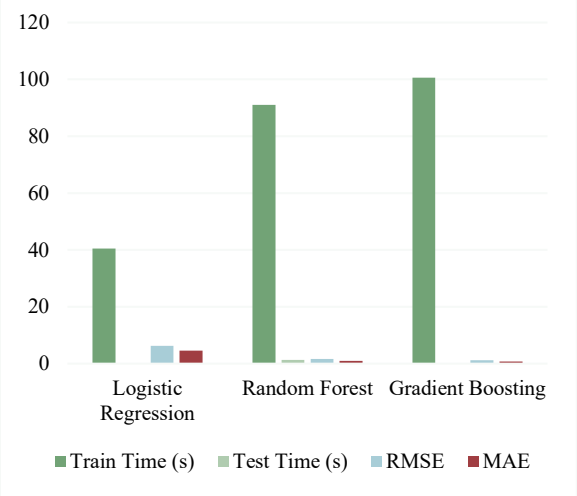
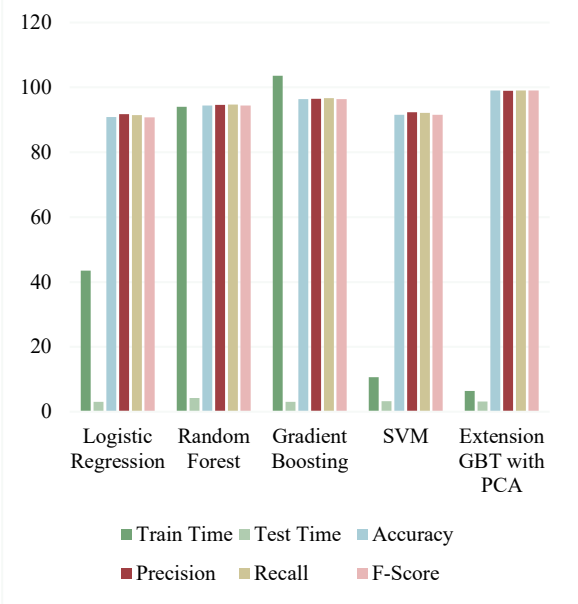


Table.2 Performance Evaluation Table - Classification

Algorithm Name	Train Time	Test Time	Accuracy	Precision	Recall	F-Score
Logistic Regression	43.48	3.08	90.82	91.76	91.41	90.81
Random Forest	94.03	4.27	94.39	94.62	94.75	94.39
Gradient Boosting	103.61	3.09	96.43	96.44	96.66	96.42
SVM	10.64	3.27	91.58	92.34	92.13	91.58
Extension GBT with PCA	**6.39**	**3.11**	**99.03**	**99.00**	**99.05**	**99.03**

Graph.3 Comparison Graph – Classification



The Comparison graph.2 uses Train Time (green) and Test Time (light green) as a model assessment measure and RMSE (blue) and MAE (red) to compare models, meaning that Gradient Boosting has the longest Train Time but performs better in measures of accuracy.

The Comparison graph.3 demonstrates the performances of the model using such measures as train time (green), test time (light green), accuracy (blue), precision (red), recall (yellow), and F-score (pink). GBT using PCA achieved the highest results on all measures.

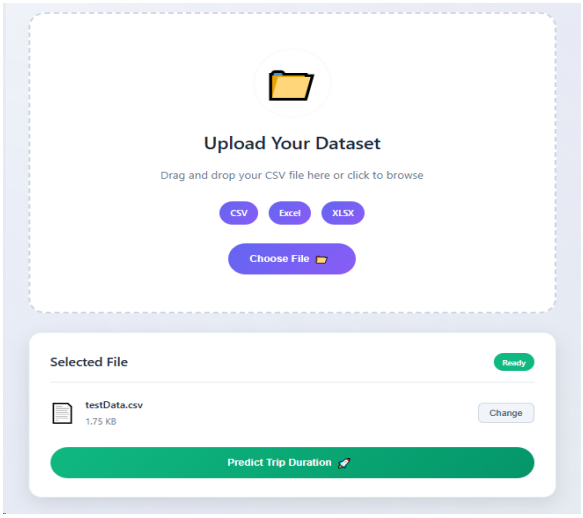


Fig .4 Upload Your Dataset

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper



Fig.5 Trip Duration Prediction Result

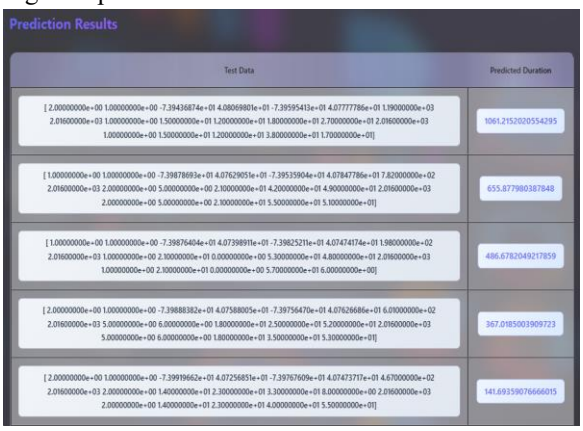


Fig.6 Prediction Result

5. CONCLUSION

The created system demonstrates the effectiveness of the distributed computing and in-memory processing by Apache Spark to execute ML algorithms on large datasets with high efficiency and accuracy. The Kaggle datasets, which are NYC Taxi Trip Duration to regression, Higgs boson to classification and Netflix Prize data to preliminary assessment, were used to apply different algorithms, including LR, RF, GBT, SVR, KNN, LR, and RMSE, MAE, training time, and testing time were used to assess the regression performance, and GBT has the lowest RMSE and MAE, and 100% accuracy of predicting the test data. GBT showed better performance in terms of Classification measures, such as Accuracy, Precision, Recall, F1-Score, Area Under the Curve (AUC), and Confusion Matrix. Adding the PCA to select the features improved the performance of the classification and PCA-enhanced GBT with 99% accuracy reduced the time to train and test. A web interface with Flask enabled real-time prediction, which shows a strong, scalable, and accurate big data analytics system.

The future scope involves the incorporation of advanced optimization techniques in features, such as genetic algorithms, as well as DL-based feature extraction, to enhance model performance. Other highly important big data processing systems, like Hadoop or Flink, can be considered in terms of scalability compared to each other. Apache Kafka can be used to stream real-time data to handle constant data feeds. The improvement of the system to be capable of handling more datasets and multi-class classification issues would expand the usage of the system, whereas a deployment of the solution in cloud environments would enhance its accessibility and scalability.

REFERENCES

[1] Singh, M., Tiwari, S. K., Swapna, G., Verma, K., Prasad, V., Patidar, V., Sharma, D. & Mewada, H. (2023). A Drug-Target Interaction Prediction Based on Supervised Probabilistic Classification. Journal of Computer Science, 19(10), 1203-1211. https://doi.org/10.3844/jcssp.2023.1203.1211

[2] Abdel Hai, A., & Forouraghi, B. (2018, June). On scalability of distributed machine learning with big data on apache spark. In International Conference on Big Data (pp. 209-219). Cham: Springer International Publishing.

[3] Aziz, K., Zaidouni, D., & Bellafkih, M. (2019). Leveraging resource management for efficient performance of Apache Spark. Journal of Big Data, 6(1), 78.

[4] Fotache, M., & Cluci, M. I. (2021, November). Big Data Performance in Private Clouds. Some Initial Findings on Apache Spark Clusters Deployed in OpenStack. In 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet) (pp. 1-6). IEEE.

[5] Junaid, M., Wagan, S. A., Qureshi, N. M. F., Nam, C. S., & Shin, D. R. (2020, October). Big data predictive analytics for apache spark using machine learning. In 2020 Global Conference on Wireless and Optical Technologies (GCWOT) (pp. 1-7). IEEE.

[6] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research

directions. Social Network Analysis and Mining, 2(3), 160. https://doi.org/10.1007/s13278-021-00735-4

[7] Shaikh, E., Mohiuddin, I., Alufaisan, Y., & Nahvi, I. (2019, November). Apache spark: A big data processing engine. In Proceedings of the 2nd IEEE Middle East North Africa Communications Conference (MENACOMM) (pp. 1–6). IEEE. https://doi.org/10.1109/MENACOMM47685.2019.8976801

[8] Kalae, U. K. (2023). Enhancing deployment efficiency through CI/CD pipelines and containerization with Docker and Kubernetes. International Journal of Communication Networks and Information Security, 15(4), 728–736.

[9] G, Viswanath., N, Madhvik., K, Bhaskar., K, Supriya. (2024). Machine-Learning-Based Cloud Intrusion Detection. International Journal of Mechanical Engineering Research and Technology, 16(9), 38-52.

[10] Quinto, B., & Quinto, B. (2020). Introduction to spark and spark MLlib. In Next-Generation Machine Learning with Spark: Covers XGBoost, LightGBM, Spark NLP, Distributed Deep Learning with Keras, and More (pp. 29–96). Apress. https://doi.org/10.1007/978-1-4842-5457-8_2

[11] Ip, R. H. L., Ang, L.-M., Seng, K. P., Broster, J. C., & Pratley, J. E. (2018). Big data and machine learning for crop protection. Computers and Electronics in Agriculture, 151, 376–383. https://doi.org/10.1016/j.compag.2018.06.008

[12] Divya, K., Bhargavi, P., & Jyothi, S. (2018). Machine learning algorithms in big data analytics. International Journal of Computer Science and Engineering, 6(1), 63–70. https://doi.org/10.26438/ijcse/v6i1.6370

[13] Aziz, K., Zaidouni, D., & Bellafkih, M. (2018, October). Big data processing using machine learning algorithms: MLlib and Mahout use case. In Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications (pp. 1–6). IEEE. https://doi.org/10.1109/SITA.2018.8660856

[14] Nair, L. R., Shetty, S. D., & Shetty, S. D. (2018). Applying spark based machine learning model on streaming big data for health status prediction. Computers & Electrical Engineering, 65, 393–399. https://doi.org/10.1016/j.compeleceng.2017.08.031

[15] G Loge, T Sunil Kumar Reddy, G Swapna, & G Viswanath. (2025). Interpretable AI for Precision Brain Tumor Prognosis: A Transparent Machine Learning Approach. In International Journal of Health Sciences and Pharmacy (IJHSP) (Vol. 9, Number 1, pp. 180–195). Zenodo. https://doi.org/10.5281/zenodo.15523628

[16] Omar, H. K., & Jumaa, A. K. (2019). Big data analysis using apache spark MLlib and Hadoop HDFS with Scala and Java. Kurdistan Journal of Applied Research, 4(1), 7–14. https://doi.org/10.24017/science.2019.1.2

[17] Hasan, Z., Xing, H. J., & Magray, M. I. (2022). Big data machine learning using Apache Spark MLlib. Mesopotamian Journal of Big Data, 2022, 1–11. https://doi.org/10.58496/MJBD/2022/001

[18] Aziz, K., Zaidouni, D., & Bellafkih, M. (2018, April). Real-time data analysis using spark and Hadoop. In Proceedings of the 4th International Conference on Optimization and Applications (ICOA) (pp. 1–6). IEEE. https://doi.org/10.1109/ICOA.2018.8370511

[19] G Ganesh, G Viswanath, G Swapna, & K Yatheendra. (2025). AI-Driven Hematological Analysis for Proactive Dengue Diagnosis. In International Journal of Health Sciences and Pharmacy (IJHSP) (Vol. 9, Number 1, pp. 196–210). Zenodo. https://doi.org/10.5281/zenodo.15541467

[20] Brahmane, A. V., & Krishna, B. C. (2021). Big data classification using deep learning and Apache Spark architecture. Neural Computing and Applications, 33(22), 15253–15266. https://doi.org/10.1007/s00521-021-05946-5