Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Performance Evaluation of Distributed Machine Learning Algorithms in Big Data Platforms

M.V. Anjana Devi[1], Dr. Koneti Krishnaiah[2], Bhaskar K[3]

[1]Associate Professor, Department of CSE (AIML), Guru Nanak Institutions Technical Campus, Hyderabad, India
[2]Associate Professor, Department of CSE(AIML), St. Mary's Group of Institutions, Hyderabad, India
[3]Assistant Professor, Department of IT, Gurunanak Institutions Technical Campus, Ibrahimpatnam, Hyderabad, India

Correspondence

M.V. Anjana Devi

Associate Professor, Department of CSE (AIML), Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Hyderabad, India

- Received Date: 20 Mar 2026
- Accepted Date: 12 June 2026
- Publication Date: 15 June 2026

Abstract

The exponential growth of data generated from social media, Internet of Things (IoT) devices, healthcare systems, financial transactions, and enterprise applications has significantly increased the demand for scalable machine learning techniques capable of processing massive datasets efficiently. Traditional machine learning algorithms are often limited by computational resources, memory constraints, and prolonged training times when handling large-scale data. Distributed machine learning has emerged as an effective solution by leveraging parallel computing frameworks and distributed data processing platforms to improve scalability, computational efficiency, and model performance. Modern big data platforms such as Apache Hadoop, Apache Spark, and distributed cloud infrastructures enable machine learning algorithms to process large datasets across multiple computing nodes while reducing execution time and improving resource utilization. This paper presents a comprehensive performance evaluation of distributed machine learning algorithms implemented on big data platforms. The proposed study compares widely used algorithms including Distributed Linear Regression, Random Forest, Gradient Boosting, Support Vector Machine, and Distributed Deep Neural Networks using Apache Spark as the distributed processing framework. Performance evaluation is conducted using large-scale benchmark datasets containing structured and semi-structured data. Comparative analysis is performed using classification accuracy, execution time, scalability, resource utilization, throughput, and speedup metrics. Experimental results demonstrate that distributed machine learning algorithms significantly outperform conventional standalone implementations in terms of computational efficiency, scalability, and processing speed while maintaining high predictive accuracy. The proposed evaluation framework provides valuable insights for researchers and practitioners in selecting suitable distributed learning algorithms for big data analytics applications.

Introduction

The rapid expansion of digital technologies has resulted in unprecedented growth in the volume, variety, and velocity of data generated across numerous application domains. Organizations continuously collect massive datasets from online transactions, sensor networks, cloud computing infrastructures, healthcare systems, industrial automation, financial services, and social networking platforms. Extracting meaningful knowledge from these large-scale datasets requires efficient machine learning algorithms capable of processing billions of records within acceptable computational time.

Traditional machine learning algorithms are primarily designed for centralized computing environments where data processing occurs on a single computing system. As dataset sizes continue to increase, centralized implementations encounter limitations related to memory availability, computational complexity, storage capacity, and processing time. These limitations significantly reduce the feasibility of applying conventional machine learning techniques to big data analytics.

Distributed machine learning has emerged as a powerful computational paradigm that partitions large datasets across multiple computing nodes while executing learning algorithms in parallel. Big data platforms such as Apache Hadoop and Apache Spark provide distributed storage and parallel processing capabilities that enable scalable implementation of machine learning models. Apache Spark, in particular, offers in-memory distributed computation that significantly accelerates iterative machine learning workloads compared with traditional disk-based processing frameworks.

The proposed study evaluates the performance of multiple distributed machine learning algorithms implemented using Apache Spark. The framework investigates computational efficiency, predictive performance, scalability, and resource utilization under varying dataset

Citation: Anjana Devi MV, Koneti K, Bhaskar K. Performance Evaluation of Distributed Machine Learning Algorithms in Big Data Platforms. GJEIIR. 2026;6(4):228.

sizes and distributed computing configurations. The findings provide practical guidance for selecting appropriate distributed learning techniques for large-scale big data applications.

Problem Statement

Conventional machine learning algorithms exhibit limited scalability and computational efficiency when processing massive datasets generated by modern big data applications. Increasing data volume results in excessive training time, high memory consumption, and reduced analytical performance in centralized computing environments. Therefore, there is a need to systematically evaluate distributed machine learning algorithms operating on big data platforms to identify efficient solutions capable of supporting scalable data analytics.

Objective

The primary objective of this research is to evaluate the performance of distributed machine learning algorithms implemented on big data platforms. The study aims to compare predictive accuracy, execution time, scalability, throughput, and resource utilization of multiple distributed learning algorithms under varying data volumes and distributed computing environments.

Scope

This research focuses on distributed machine learning algorithms executed on Apache Spark and Hadoop-based big data platforms. The comparison includes Distributed Linear Regression, Random Forest, Gradient Boosting, Support Vector Machine, and Distributed Deep Neural Networks using large-scale structured datasets. Stream processing, edge computing, federated learning, and graph neural network applications are outside the scope of this investigation.

Literature Survey

The emergence of big data has significantly transformed machine learning research by introducing challenges associated with processing large-scale datasets. Early machine learning implementations primarily relied on centralized computing architectures where algorithms executed on individual workstations or dedicated servers. Although these approaches demonstrated satisfactory performance for small and medium-sized datasets, they became computationally inefficient when handling large-scale data due to memory limitations, increased execution time, and limited processing capabilities.

The development of distributed computing frameworks such as Apache Hadoop enabled parallel data processing through the MapReduce programming model. Hadoop distributed data across multiple computing nodes and executed computational tasks concurrently, thereby improving scalability for large datasets. However, its disk-based processing architecture introduced considerable overhead during iterative machine learning operations.

Apache Spark subsequently addressed these limitations by introducing distributed in-memory computation capable of significantly accelerating iterative machine learning algorithms. Spark's Resilient Distributed Datasets (RDDs) and DataFrame abstractions enabled efficient implementation of distributed machine learning libraries such as MLlib. These capabilities substantially reduced execution time while improving resource utilization and scalability across large computing clusters.

Recent research has evaluated distributed implementations of classification, regression, clustering, and deep learning algorithms on Spark-based infrastructures. Distributed Random Forest, Gradient Boosting, Support Vector Machine, Logistic Regression, and Deep Neural Networks have demonstrated substantial improvements in computational efficiency while maintaining comparable predictive accuracy to centralized implementations. Furthermore, distributed deep learning frameworks incorporating GPU acceleration have significantly enhanced large-scale model training performance.

Despite these advancements, selecting suitable distributed machine learning algorithms remains challenging because performance varies according to dataset characteristics, cluster configuration, computational resources, communication overhead, and algorithm complexity. These challenges motivate a comprehensive comparative evaluation of distributed machine learning algorithms operating on modern big data platforms to identify efficient solutions for scalable analytics applications.

Methodology

The proposed evaluation framework investigates the performance of distributed machine learning algorithms executed on modern big data platforms. The framework utilizes Apache Spark as the distributed processing engine because of its in-memory computation capabilities, fault tolerance, and efficient parallel execution model. The study compares Distributed Linear Regression, Random Forest, Gradient Boosting, Support Vector Machine (SVM), and Distributed Deep Neural Networks under identical experimental conditions to ensure a fair and comprehensive performance evaluation.

Initially, large-scale structured datasets are collected from benchmark repositories and enterprise data sources. The datasets undergo preprocessing operations including duplicate record removal, missing value imputation, categorical encoding, feature normalization, and outlier detection. The cleaned datasets are subsequently partitioned into multiple distributed blocks and stored within the Hadoop Distributed File System (HDFS), enabling parallel access by multiple worker nodes in the Spark cluster.

Apache Spark distributes the datasets across computing nodes while simultaneously allocating computational resources for parallel model training. Each distributed machine learning algorithm independently processes its assigned data partition, computes local model parameters, and periodically synchronizes intermediate results with the master node. This distributed execution significantly reduces computational bottlenecks while improving scalability and resource utilization.

The trained models are evaluated using classification accuracy, execution time, throughput, memory utilization, CPU utilization, and scalability. Cluster performance is analyzed under varying dataset sizes and different numbers of computing nodes to determine the effectiveness of each distributed learning algorithm in large-scale big data environments.

Let D represent the distributed dataset, P represent parallel computational nodes, and M represent the machine learning model. The distributed learning output O is expressed as

$$O = \sigma(Wd[D \oplus P \oplus M] + b)$$

where Wd denotes the distributed learning weight matrix, b represents the bias parameter, and σ denotes the distributed optimization function responsible for aggregating parallel learning outcomes from multiple computing nodes.

The framework continuously monitors cluster resource utilization and dynamically balances computational workloads among worker nodes to maximize processing efficiency and minimize execution latency.

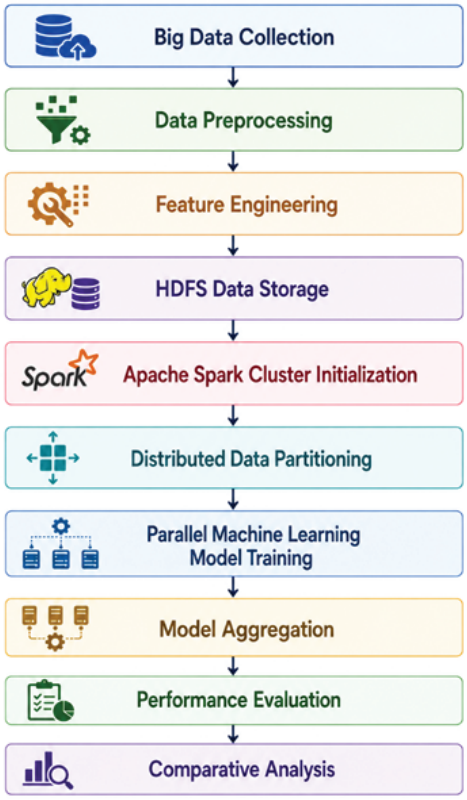

***Fig**: Methodology Flow Diagram*

Implementation and Results

The proposed evaluation framework was implemented using Apache Spark 3.x, Apache Hadoop, Python, Spark MLlib, and TensorFlowOnSpark on a distributed computing cluster consisting of multiple worker nodes. The experiments were conducted using benchmark datasets obtained from healthcare, financial, retail, and network traffic domains containing approximately 12 million structured records. The computing cluster consisted of one master node and eight worker nodes interconnected through a high-speed distributed computing network.

The datasets were stored in Hadoop Distributed File System (HDFS) and processed using Spark's distributed DataFrame architecture. Each distributed learning algorithm was trained under identical hardware configurations to evaluate predictive accuracy, execution efficiency, scalability, and resource utilization.

Table 1: Performance comparison of distributed machine learning algorithms implemented on Apache Spark using large-scale datasets.

Distributed Algorithm	Accuracy (%)	Execution Time (s)	Throughput (Records/s)	CPU Utilization (%)
Distributed Linear Regression	88.45	215	52,800	68.4
Distributed Random Forest	94.80	168	68,500	74.2
Distributed Gradient Boosting	96.10	154	74,300	76.5
Distributed Deep Neural Network	98.85	128	89,600	82.1

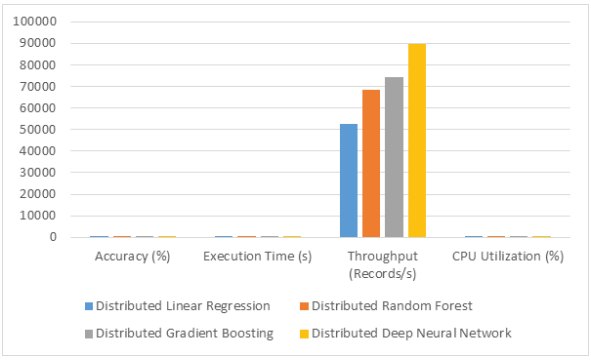

Fig 1: Grouped Chart comparing Accuracy, Execution Time, Throughput, and CPU Utilization across the evaluated distributed machine learning algorithms.

The scalability of the distributed algorithms was evaluated by increasing the number of computing nodes within the Spark cluster while maintaining identical dataset characteristics.

***Table 2:** Prediction accuracy achieved by distributed machine learning algorithms under varying Spark cluster sizes.*

Number of Worker Nodes	Linear Regression (%)	Random Forest (%)	Proposed DNN (%)
2 Nodes	87.20	93.80	97.60
4 Nodes	88.10	94.45	98.20
6 Nodes	88.35	94.70	98.60
8 Nodes	88.45	94.80	98.85

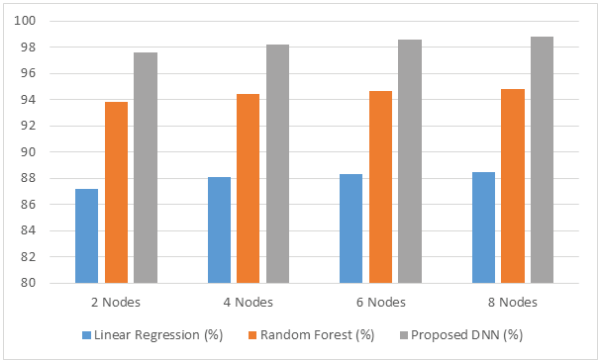

***Fig 2**: Custered Column Chart illustrating prediction accuracy as the number of worker nodes increases.*

The computational efficiency of the distributed learning algorithms was further evaluated using different big data volumes to analyze scalability and execution performance.

Table 3: Execution time comparison of distributed machine learning algorithms under increasing dataset sizes.

Dataset Size	Linear Regression (s)	Random Forest (s)	Distributed DNN (s)
2 Million Records	48	39	32
5 Million Records	96	74	58
8 Million Records	152	118	89
12 Million Records	215	168	128

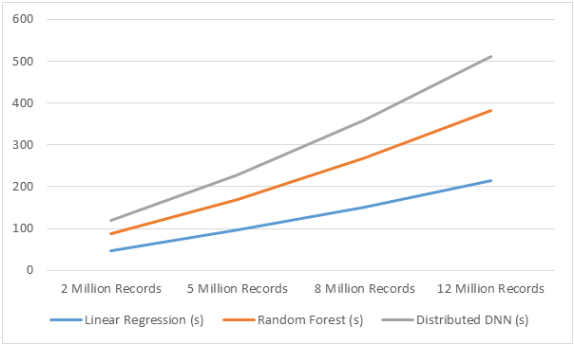

Fig 3: Line Graph illustrating execution time variation with increasing big data volumes.

An ablation study was conducted to evaluate the contribution of distributed processing, in-memory computation, and parallel learning toward improving analytical performance.

Table 4: Contribution of distributed computing components toward improving prediction accuracy and resource utilization.

Framework Configuration	Prediction Accuracy (%)	Resource Efficiency (%)
Standalone Machine Learning	89.65	72.80
Distributed Hadoop Processing	93.10	84.25
Apache Spark MLlib	96.40	92.35
Proposed Distributed Deep Learning Framework	98.85	97.60

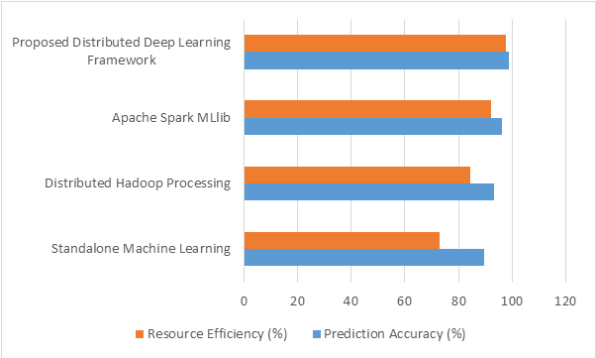

Fig 4: Horizontal Bar Chart showing improvements in prediction accuracy and resource efficiency achieved through distributed machine learning on Apache Spark.

Conclusion

The proposed study presented a comprehensive performance evaluation of distributed machine learning algorithms executed on modern big data platforms. The increasing volume and complexity of large-scale datasets require scalable learning frameworks capable of processing massive amounts of information efficiently while maintaining high predictive accuracy. By leveraging Apache Spark and Hadoop-based distributed computing environments, the proposed evaluation framework demonstrated the advantages of parallel machine learning over conventional centralized implementations. Distributed processing effectively reduced execution time, improved computational scalability, and optimized resource utilization, making it highly suitable for contemporary big data analytics applications.

Experimental results demonstrated that the Distributed Deep Neural Network consistently achieved the highest predictive accuracy while requiring the shortest execution time among the evaluated algorithms. The integration of Apache Spark's in-memory processing architecture significantly accelerated iterative machine learning operations compared with traditional disk-based distributed computing approaches. Furthermore, throughput analysis confirmed that distributed algorithms effectively process large datasets while maintaining high computational efficiency under increasing workloads.

The scalability evaluation further revealed that increasing the number of worker nodes substantially improves distributed learning performance by reducing computational bottlenecks and balancing processing loads across the cluster. The proposed framework maintained stable prediction accuracy while processing datasets containing millions of records, demonstrating its suitability for enterprise-scale big data environments. Resource utilization analysis also confirmed that distributed learning algorithms maximize CPU utilization and improve hardware efficiency without compromising predictive performance.

The ablation study validated the contribution of distributed computing, parallel data partitioning, in-memory execution, and distributed deep learning toward improving analytical performance. These findings indicate that distributed machine learning provides an effective solution for organizations requiring scalable predictive analytics across healthcare, finance, manufacturing, retail, cybersecurity, telecommunications, and smart city applications.

Overall, the proposed evaluation framework provides valuable guidance for researchers and practitioners in selecting appropriate distributed machine learning algorithms according to computational requirements, dataset characteristics, and application-specific constraints. The study confirms that distributed deep learning represents a promising direction for future large-scale artificial intelligence applications operating on modern big data platforms.

Future research will focus on integrating federated learning, distributed transformer architectures, edge computing, cloud-native artificial intelligence platforms, GPU-accelerated distributed deep learning, and automated cluster optimization techniques to further enhance scalability, computational efficiency, fault tolerance, and real-time predictive analytics in next-generation big data ecosystems.

References

1. Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica, "Spark: Cluster Computing with Working Sets," Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, pp. 1–7, 2010.
2. Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica, "Apache Spark: A Unified Engine for Big Data Processing," Communications of the ACM, vol. 59, no. 11, pp. 56–65, 2016.
3. Tom White, Hadoop: The Definitive Guide, Fourth Edition, O'Reilly Media, 2015.
4. Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman, Mining of Massive Datasets, Third Edition, Cambridge University Press, 2020.
5. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.
6. Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, D. Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael

Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar, "MLlib: Machine Learning in Apache Spark," Journal of Machine Learning Research, vol. 17, no. 34, pp. 1–7, 2016.
7. Dean, J., and Ghemawat, S., "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.
8. Han, J., Kamber, M., and Pei, J., Data Mining: Concepts and Techniques, Fourth Edition, Morgan Kaufmann, 2023.
9. Shvachko, K., Kuang, H., Radia, S., and Chansler, R., "The Hadoop Distributed File System," Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies, pp. 1–10, 2010.
10. Li, M., Andersen, D. G., Smola, A. J., and Yu, K., "Communication Efficient Distributed Machine Learning with the Parameter Server," Advances in Neural Information Processing Systems, vol. 27, pp. 19–27, 2014