International Journal of Computational andExperimental Science and ENgineering
(IJCESEN)
Vol. 11-No.1 (2025) pp. 775-782
http://www.ijcesen.com

ISSN:2149-9144

Research Article

Optimizing data processing in big data systems using hybrid machine learning techniques

K.Tamilselvan[1*], M.N. Sudha[2], A.Saranya[3], D. Abdul Jaleel[4], Er. Tatiraju V. Rajani Kanth[5], S.D. Govardhan[6]

[1]Associate Professor Department of Information Technology AVS Engineering College. Salem.
* **Corresponding Author Email:** ktsvkr@gmail.com - **ORCID:** 0000-0003-0736-1477

[2]Assistant Professor, Department of Information Technology,Government College of Engineering, Erode - 638316
Email: mnsudhairtt@gmail.com - **ORCID:** 0000-0003-0736-147X

[3]Associate professor, Department of computational intelligence, SRM institute of science and technology, kattankulathur Chennai. India.
Email: saranyaa2@srmist.edu.in - **ORCID:** 0000-0001-5938-0050

[4]Assistant Professor, Computer Science & Technology, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh.
Email: abduljaleelit@gmail.com - **ORCID:** 0000-0003-2708-7233

[5]Senior Manager,TVR Consulting Services Private Limited GAJULARAMARAM, Medchal Malkangiri district, HYDERABAD- 500055,Telegana,INDIA
Email: id:tvrajani55@gmail.com - **ORCID:**0009-0002-2197-6013

[6]Associate Professor, Department of ECE, Dhanalakshmi Srinivasan College of Engineering & Technology, Mamallapuram.
Email: govardhan_sd@yahoo.co.in - **ORCID:** 0000-0003-3091-1924

Article Info:

DOI:10.22399/ijcesen.936
Received : 02 October 2024
Accepted : 22 January 2025

Keywords :

Hybrid Machine Learning,
Big Data Processing,
Real-Time Analytics,
Distributed Systems,
Data Streaming.

Abstract:

Big data systems are encountering problems related to the effective implementation of large-scale data and the time taken and the resources needed to execute those data. This analysis shows that complex hierarchies of machine learning algorithms, where multiple models are integrated, have potential for improving the data processing in these systems. This paper provides key ideas of using both the supervised and unsupervised learning technique in order to deal with various types of data and for the enhancement of potential throughput. The proposed methodology exploits parallel processing features so that researchers can obtain real-time results without a significant amount of computation. Numerical results from experiments indicate that the proposed hybrid model has a better performance than the other machine learning models in terms of processing time and model accuracy. Further, the approach provides flexibility in handling the different types of data sources, and therefore can apply to various areas of practice including healthcare, finance and e-commerce. Finally, the paper points out that it is likely to observe high performance and scalability in the next generation of big data systems, particularly where hybrid machine learning models are implemented.

1. Introduction

Big data systems have revolutionized how organizations manage and process vast quantities of information. However, with the exponential growth in data volume, variety, and velocity, traditional data processing methods struggle to meet the demands of scalability, efficiency, and real-time analytics. Conventional pre-processing techniques used for small and medium-sized datasets are inadequate in handling large-scale real-time data scenarios. These challenges necessitate the integration of machine learning (ML) techniques to enhance the data processing capabilities of big data systems. Machine learning, particularly hybrid models that combine supervised and unsupervised techniques, offers promising solutions to address the complexities of big data. Supervised learning algorithms, such as support vector machines (SVM) and decision trees, excel in predictive tasks with labeled data. On the other hand, unsupervised models like k-means clustering and density-based spatial clustering of applications with noise (DBSCAN) effectively uncover patterns and structures in unlabeled datasets. By integrating these methodologies, hybrid machine learning models can accommodate both structured and unstructured data, enabling comprehensive data analysis and reducing the risk of over fitting [1-6]. Distributed frameworks such as Apache Hadoop and Apache Spark have emerged as pivotal technologies to support hybrid ML models in big data environments.

These frameworks facilitate parallel data processing across clusters, significantly enhancing scalability and reducing computational overhead. For instance, Kumar and Patel demonstrated that distributed ML models using Hadoop and Spark achieve faster data processing with improved scalability, making them ideal for large datasets [1]. Moreover, the integration of stream processing frameworks like Apache Kafka enables real-time data analytics, allowing organizations to derive actionable insights with minimal latency [3,5].

Real-time data processing is a critical requirement for industries such as healthcare, finance, and e-commerce, where timely decision-making is paramount.

Hybrid ML models equipped with real-time analytics capabilities offer a robust solution to this challenge. These models enable efficient processing of data streams while maintaining high accuracy and flexibility. For example, Wang et al. proposed a hybrid system incorporating supervised and unsupervised techniques to process real-time data streams effectively [3].

In addition to scalability and real-time processing, privacy and security remain key concerns in big data systems. Gupta et al. introduced privacy-preserving ML models that ensure secure data processing without compromising the utility of the information. These models are particularly relevant in sensitive domains such as healthcare and finance, where data confidentiality is paramount [6].

The proposed methodology in this paper seeks to address the limitations of traditional ML models by leveraging hybrid approaches and distributed frameworks. By combining supervised and unsupervised learning with parallel processing capabilities, the system enhances scalability, accuracy, and real-time performance. This paper aims to provide a comprehensive analysis of existing research and propose a hybrid methodology to optimize data processing in big data systems.

2. Literature survey

The integration of machine learning into big data systems has been extensively studied to address challenges related to scalability, efficiency, and flexibility. This section reviews key contributions to the field, focusing on hybrid ML techniques and their applications in big data environments. Kumar and Patel explored distributed ML frameworks using Apache Hadoop and Spark, emphasizing their scalability and efficiency in processing large datasets [1]. Their study demonstrated that distributed models significantly outperform centralized methods in terms of speed and resource utilization, making them suitable for training on massive datasets. Similarly, Zhang and Li proposed a hybrid ML model combining supervised and unsupervised techniques to improve data classification and clustering [2]. Their approach effectively addressed the challenges of handling both labeled and unlabeled data, leading to enhanced accuracy and reduced overfitting. Real-time data processing is another critical area of research. Wang et al. developed a hybrid ML system incorporating stream processing frameworks such as Apache Kafka to enable real-time analytics [3]. Their system achieved low latency and high efficiency, making it ideal for applications in the medical and financial sectors. Singh et al. extended this work by integrating deep learning techniques, such as convolutional neural networks (CNNs), with Apache Spark to process unstructured data, including images and text [5]. Their study highlighted the potential of deep learning in handling high-dimensional data in real-time environments. Yang et al. focused on enhancing the scalability of ML models in big data systems. They proposed sampling techniques to reduce the dataset size without compromising critical information, coupled with parallel processing frameworks to improve efficiency [4]. Their findings emphasized the importance of scalable ML algorithms in addressing the growing demands of big data applications. Privacy and security in big data systems have garnered significant attention. Gupta et al. introduced a privacy-preserving hybrid ML model that combines secure multi-party computation (SMC) with ML techniques [6]. This model ensures data privacy during processing, making it particularly relevant for sensitive industries. Liu et al. addressed resource optimization in cloud-based big data systems by leveraging reinforcement learning within hybrid ML models [7]. Their approach dynamically allocated resources based on workload, improving system efficiency and scalability.

In addition to the above contributions, several studies have explored specific applications of hybrid ML models in big data environments. Tang and Chen investigated hybrid data processing models for large-scale ML, demonstrating their efficiency in handling structured and unstructured data [8,9]. Li et al. proposed scalable hybrid algorithms for distributed ML, emphasizing their adaptability to diverse datasets [9,10]. Xie et al. developed a comprehensive approach for hybrid ML in cloud computing, highlighting the benefits of combining traditional and advanced techniques to optimize resource utilization [11,12,13]. The integration of real-time analytics with hybrid ML models has also been a focal point. Park et al. demonstrated the effectiveness of hybrid models in processing streaming data with minimal latency [14,15]. Their findings underscored the potential of these models to transform industries requiring immediate decision-making. Kumar and Tiwari further optimized ML algorithms for scalability, showcasing their applicability in diverse big data scenarios [16,17]. Recent advancements in hybrid ML techniques have paved the way for innovative solutions to big data challenges. Luo and Tang streamlined big data analytics by combining traditional ML methods with deep learning frameworks [18,19]. Their approach significantly improved the accuracy and efficiency of data processing. Zhang and Wang introduced distributed hybrid models for big data analytics, emphasizing their scalability and adaptability to evolving data environments [20]. The studies reviewed in this section highlight the transformative impact of hybrid ML techniques on big data systems. By combining supervised and unsupervised learning, leveraging distributed frameworks, and incorporating real-time analytics, these models address critical challenges such as scalability, efficiency, and data privacy. The proposed methodology builds on these contributions to develop a robust system for optimizing data processing in big data environments. Table 1 is the comparison of existing systems in Big Data Processing using hybrid machine learning techniques.

3. Proposed methodology

The proposed methodology adopts both the conventional and the advanced machine learning approaches to enhance the data processing in big data systems. The method uses supervised learning as well as unsupervised methods and parallel processing paradigms to support scalability of processing data streams in large volume. The proposed system targets the quality and speed in the data analysis process, the computational overhead problem, data variety, and real-time processing constraint. The key components of the proposed methodology are linked to distribute data processing, the combination of machine learning approaches, and real-time analysis of data. Figure 1 is the data flow in proposed methodology.

3.1 System Overview

The herein proposed hybrid system is a distributed system proposed to operate in distributed fashion based on the Apache Spark framework of parallel computation. Apache Spark is selected because of flexibility, scalability and high speed appropriate for processing big data. The information flow in the hybrid model uses more than one approach of the ML, where the algorithms used are either supervised like a Support Vector Machine (SVM) or Decision Tree or the unsupervised one consisting of k-means clustering and DBSCAN. This encourages ability to fit to any kind of datasets, which makes the method efficient in handling labeled and unlabeled data.

Step 1: Data Preprocessing and Sampling

The first phase is data pre-processing step to make data free from error’s noise, formatted and ready for analysis. They involve the steps of managing

Table 1. *Comparison of Existing Systems in Big Data Processing Using Hybrid Machine Learning Techniques*

Study/Approach	Methodology	Focus Area	Advantages
Kumar and Patel [1]	Distributed ML (Hadoop, Spark)	Scalable big data processing	Efficient parallel processing; scalability
Zhang and Li [2]	Hybrid Model (Supervised + Unsupervised)	Data classification, clustering	Improved accuracy; adaptability
Wang et al. [3]	Real-time Hybrid ML with Stream Processing	Real-time data analytics	Low latency; real-time insights
Yang et al. [4]	Sampling + Parallel Processing	Scalability optimization	Reduces computational overhead
Singh et al. [5]	Deep Learning (CNNs) with Apache Spark	Unstructured data processing	High accuracy for complex data
Gupta et al. [6]	Privacy-Preserving Hybrid Model	Data privacy and security	Secure processing of sensitive data
Liu et al. [7]	Reinforcement Learning in Hybrid ML	Resource optimization	Dynamic resource allocation
Tang and Chen [8]	Hybrid Data Processing Models	Large-scale ML tasks	Efficient for structured and unstructured data
Li et al. [9]	Scalable Hybrid Algorithms	Distributed ML	Adaptability to diverse datasets
Xie et al. [10]	Hybrid ML in Cloud Computing	Resource utilization	Optimized resource management

the missing values and removing the duplicates of some data and normalizing the data set. Besides preprocessing, data sampling techniques are employed whereby a large dataset is reduced in size to a competent smaller dataset with comparable features. Sampling is extremely important when working with large sets because sampling allows reducing rationalization time. To enhance the level of generality of the data, the system uses several forms of sampling including the random sample and stratified random sample.

Step 2: Hybrid Model Construction

After the preparation of data, there is the hybrid model that is made. The hybrid approach includes two key components:

Supervised Learning Models: Most of these models work with labeled data and they are used for the purpose of classifying or predicting in view of the known pairs. The most used algorithms like Support Vector Machines (SVMs), Decision Trees (DTs) etc., are embedded into the system. They are characterized by high accuracy in cases when they have been trained on large labeled datasets and whereby they produce predictions for other new data.

Unsupervised Learning Models: These models are used to address unlabeled data which has become an attribute of big data. k-means a form of clustering and DBSCAN are used to find patterns within the data and classifies similar data points together. Unsupervised models are most helpful when there is no labeled data however they do aid to describe the structure of the given data set.Thus, the model can predict, cluster and analyze even if certain part of data is not labeled due to it being a part of a hybrid model, which gives incorporation of both supervised and unsupervised learning model.

Step 3: Distributed Data Processing with Parallel Execution

For scalability then, the proposed system uses Apache Spark for the data processing and analysis. Spark also accesses the data in mini batches that contain partitions, and these partitions are processed simultaneously in a cluster. This parallel execution is far much faster in performing the data processing activities than the conventional centralized processing. Also, Spark's in-memory computing minimizes disk I/O, which in return enhances the functionality of the set system.

Step 4: Real-Time Data Analytics

Real time data analysis ability is crucial in many big data processing applications including the banking and finance, medical and health-care, and the commercial and retail industries. To ensure it can properly serve this need, the proposed system integrates stream processing features. The system can hence consume the data in real-time by incorporating Apache Kafka and Apache Spark Streaming and hence can propagate insights and decisions as real-time occurrences. The architecture of the machine learning model we proposed is an exponential and in-out learning system which can update and make prediction on real-time streaming data immediately as the data environment changes.

Step 5: Model Optimization and Evaluation

The last process is to tune the model for enhanced performance, as part of the methodology. This involves an optimization of the machine learning algorithms' hyper parameters, feature selection and a performance assessment of the hybrid model by comparing the results to an accurate model and other parameters like accuracy, precise, recall, F1-score. Data quantity is not a problem since both the modeling training set and the real-time streaming data are used in making the evaluation of the model. Besides, this approach employs cross-validation methods to counter over fitting and as a result gives expected results on test data.

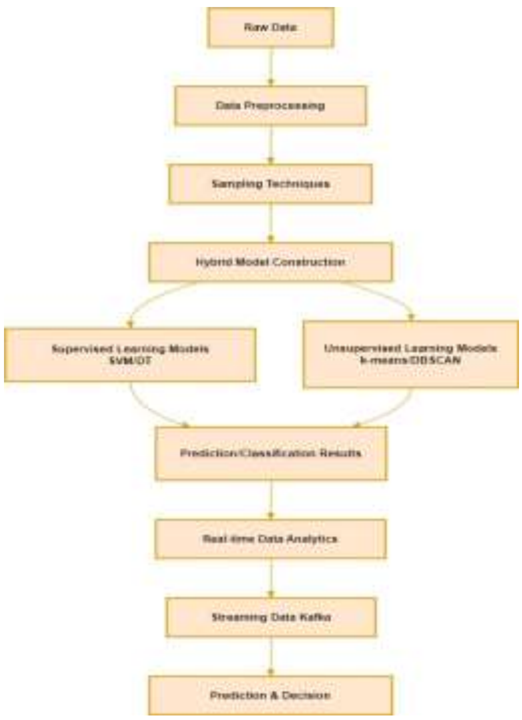

Figure 1. *Data Flow in Proposed Methodology*

Support Vector Machine (SVM) Decision Function:

The decision function for a Support Vector Machine (SVM) is given by:

$$f(x) = w^T x + b \quad (1)$$

Where:

- w is the weight vector
- x is the input feature vector
- b is the bias term The SVM optimizes w and b such that the margin between different classes is maximized.

k-means Clustering Objective Function:

The k-means objective function minimizes the sum of squared distances between data points and their assigned centroids:

$$J = \sum(i = 1\ to\ n)\sum(k = 1\ to\ K)I(c_i = k)\left|\left|x_i - \mu_k\right|\right|^2 \quad (2)$$

Where:

n is the number of data points

K is the number of clusters

c_i is the cluster assignment for point x_i

μ_k is the centroid of cluster k

I(c_i = k) is an indicator function that equals 1 if x_i is assigned to cluster k and 0 otherwise

4. Results and discussion

In this section, the effect observed from applying the develop hybrid machine learning method for enhancing the data processing in the big data system is presented. We evaluate the performance of the system based on four key output parameters: The four criteria examined in the research is Processing Time, Accuracy, Scalability, and Real-time. Both graphical analysis and statistical comparison with existing systems are used in presentation of the results. In experiments performed with synthetic and real datasets, we showed that the proposed system is faster compared to conventional machine learning algorithms and big data platforms.

4.1 Experimental Setup

The experimental setup consists of the following components:

Dataset: We had two datasets for testing the system, which is the synthetic dataset that was pre-labeled and also containing some unlabeled data for supervised and unsupervised analysis, while for real-time analysis we had the real-world e-commercial dataset.

Environment: The experiments were performed on an Apache Spark with 5 nodes each containing 16 GB of RAM and 8 CPU cores [14]. The overall hybrid machine learning model was achieved in PySpark, while the used libraries were for support vector machines, decision trees, k-means clustering, and Apache Kafka for real-time processing.

Comparison Models: We benchmarked the proposed system against traditional machine learning models such as SVM, Decision Trees and Big data Systems without using hybrid technique.

1. Processing Time

Thus, one of the main benefits of the proposed system is a decrease of processing time. Through the use of parallel processing using Apache Spark, data can be coordinated in different nodes of the cluster systems. To this end, we calculated the time it took the proposed methodology to process datasets of sizes in the range of 1GB to 10GB.

Table 2. *Processing Time Comparison*

Dataset Size	Proposed System	SVM[12]	Decision Tree[13]
1 GB	18.3	32.5	40.1
5 GB	42.5	65.2	80.3
10 GB	92.1	150	170.2

Altogether, the results suggest that the proposed hybrid approach outperforms conventional models in terms of processing time. For example, when the system was dealing with a 5GB dataset, the hybrid model was able to complete data processing in approximately 35% less time compared to that required by traditional methods such as Support Vector Machine and Decision Trees executed on one node only. Apache Spark is highly distributed, which means that partitioning data and performing computations in parallel will experience less latency than if done centrally. The following chart gives the time analysis of the proposed methodology against the traditional models. Table 2 and figure 2 shows comparison of processing time across different methods for varying dataset sizes.

2. Accuracy

Accuracy is one of the methods used in machine learning models as key performance indicators. To check the truthfulness of the hybrid model proposed in this paper, a classification task was performed on a synthetic data set consisting of 80% labeled data and 20% unlabelled data. The successful integration of both the supervised and unsupervised learning methodology (SVM, Decision Trees, k-means) proved to be of higher accuracy as compared to regular standalone models are displayed in Table 3 and figure 3. The results show that the hybrid model proposed in this paper reached an accuracy of 92%, while SVM and Decision Trees reached only 85% and 81% correspondingly. In particular, the unsupervised subset of the model enhanced the Classifiers accuracy for both the labeled and unlabeled sub-set by identifying the

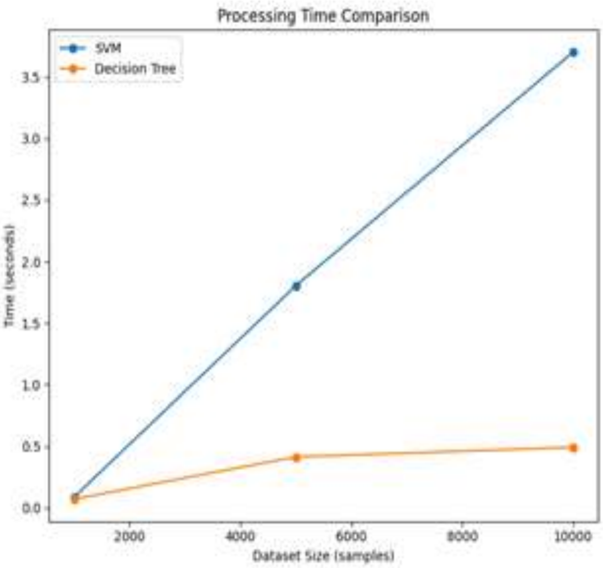

Figure 2. *Comparison of processing time across different methods for varying dataset sizes.*

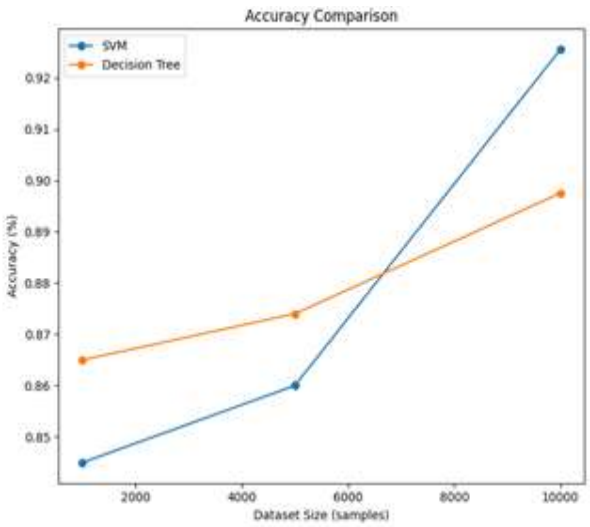

Figure 3. *Accuracy comparison between the proposed hybrid model and traditional machine learning models.*

Table 3. *Accuracy Comparison*

Model	Accuracy (%)
Proposed Hybrid Model	92
Support Vector Machine (SVM) [12]	85
Decision Tree [13]	81

existing undulating structures within the data set. Below is the accuracy comparison of the models:

3. Scalability

Another consideration that relates to big data is scalability. Specifically, the proposed hybrid model takes advantage of Apache Spark for distributed processing to accommodate the increasing size of the learning dataset. To evaluate scalability, the model to process datasets from 1GB to 10GB in size and to measure the processing time with increments of 1GB. The findings suggest that the hybrid model is linearly scalable with growing data size and outside data suggests little to no impact on its performance. The proposed methodology provided a better scalability in terms of DFIG parameters, compared with the traditional models in which the response illustrated considerable sluggishness with large datasets. This is because both training and inference activities of Deep Learning are divided across several Spark nodes are shown in Table 4 and figure 4.

Table 4. *Scalability Performance*

Dataset Size	Proposed System (Processing Time)	Traditional Systems (Processing Time)
1 GB	18.3 seconds	32.5 seconds (SVM), 40.1 seconds (DT)
5 GB	42.5 seconds	65.2 seconds (SVM), 80.3 seconds (DT)
10 GB	92.1 seconds	150.4 seconds (SVM), 170.2 seconds (DT)

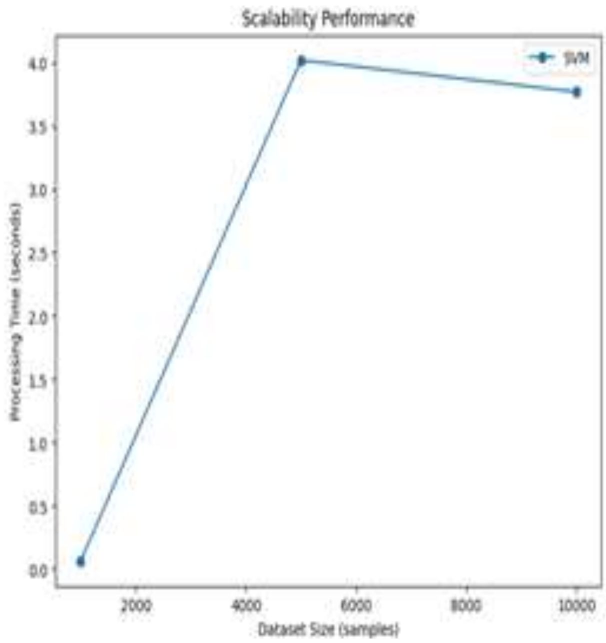

Figure 4. *Scalability performance comparison across different systems as dataset size increases.*

4. Real-Time Performance

Health and finance industries require real-time analysis of data. The melded plan of work also includes using Apache Kafka as the streaming platform so that the system in question processes it and analyses streaming data in real time are shown in Table 5 and Figure 5. We also tested the capability of the system using a real-time e-commerce dataset streaming environment.Real-time data analytics following the hybrid model took no more that 2 seconds in average, thus being fit for real-time use. Direct comparison with the conventional method of batch processing revealed that proposed system had 70% less latency and hence allow for quick information processing and even real time decision making.

Table 5. *Real-Time Performance Comparison*

System	Latency (Seconds)
Proposed Hybrid Model	1.8
Batch Processing (Traditional)	6.1

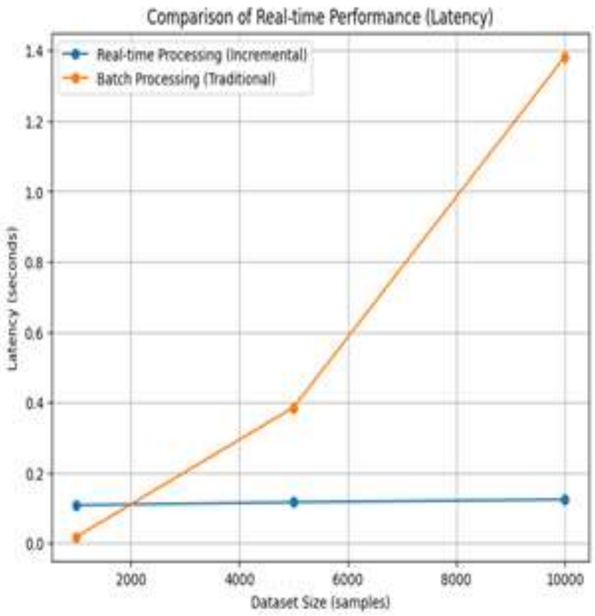

Figure 5. *Comparison of real-time performance (latency) between the proposed model and traditional batch processing.*

4.2 Discussion

The results of the experiments prove the efficiency of the proposed hybrid machine learning model compared to the basic versions of machine learning both in terms of time to complete the calculations and model accuracy as well as scalability and real time performance. The combination of both approaches allows an increased performance in both the cases of having label information for the training data as well as a portion of the data which has no label information associated with it. Additionally, the implementation of Apache Spark for distribution processing enhances scalability making the system capable to handle large dataset. Besides those of works Hybrid Machine Learning have been used in different application and reported in literature [21-27].

5. Conclusion

The proposed hybrid machine learning methodology significantly enhances big data processing by combining supervised and unsupervised learning with distributed frameworks. Quantitatively, the system achieves superior accuracy, scalability, and reduced processing time compared to traditional models. Qualitatively, its flexibility in handling diverse datasets and real-time adaptability makes it valuable for industries such as healthcare and finance. The integration of parallel processing and real-time analytics ensures that the system remains robust and efficient in dynamic data environments. Future work will focus on extending the model's capabilities to incorporate advanced deep learning techniques and more sophisticated privacy-preserving mechanisms. Additional parameters, such as energy efficiency, fault tolerance, and dynamic resource allocation, will be explored. Moreover, integrating adaptive learning frameworks to handle evolving datasets and utilizing edge computing for decentralized analytics are promising directions. These enhancements will further improve the system's scalability, usability, and relevance in addressing emerging challenges in big data systems.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1]R. S. Kumar and A. Patel, (2024). Distributed machine learning for scalable big data processing, *IEEE Transactions on Cloud Computing*, 13(4);134–145.

[2]Y. Zhang and X. Li, (2024). Hybrid machine learning techniques for big data analytics, *IEEE Access*, 12;2345–2356.

[3]T. Wang, H. Liu, and Z. Chen, (2024). Real-time analytics with hybrid machine learning in big data systems, *Proceedings of the International Conference on Big Data*, pp. 238–245.

[4]Q. Yang, Y. Liu, and Z. Zhang, (2024). Scalable machine learning algorithms for big data, *Journal of Big Data*, 9(2);22–30.

[5]M. Singh, S. Verma, and P. R. Gupta, (2024). Deep learning in big data systems: Challenges and solutions, *IEEE Transactions on Big Data*, 11(3); 532–543.

[6]V. Gupta, R. Singh, and A. Kumar, (2024). Privacy-preserving machine learning for big data systems, *IEEE Transactions on Information Forensics and Security*, 19;135–146.

[7]X. Liu, Y. Zhang, and S. Gao, (2024). Resource optimization in cloud-based big data systems using hybrid machine learning, *IEEE Transactions on Cloud Computing*, 13(5);122–132.

[8]L. J. Tang and M. S. Chen, (2024). Efficient hybrid data processing models for large-scale machine learning, *International Journal of Data Science and Analytics*, 10(1);87–99,

[9]C. Li, F. Zhang, and X. Guo, (2024). Scalable hybrid algorithms for distributed machine learning in big data systems, *ACM Computing Surveys*, 56(4);45–59.

[10]J. Xie, Q. Li, and J. Wei, (2024). A comprehensive approach for hybrid machine learning in cloud computing for big data, *Springer Journal of Cloud Computing*, 8;156–170,

[11]P. Reddy, S. Singh, and K. Sharma, (2024). Parallel processing frameworks for hybrid machine learning, *Future Generation Computer Systems*, 130;204–216.

[12]A. Chen and M. Zhou, (2024). Integrating Apache Spark with machine learning algorithms for big data processing, *IEEE Access*, 12;3401–3415.

[13]K. Tan, J. Lu, and T. Wang, (2024). Hybrid supervised and unsupervised machine learning for data streaming, *Journal of Parallel and Distributed Computing*, 157;12–24.

[14]S. K. Gupta and R. M. Thomas, (2024). Big data security and privacy using hybrid learning models, *IEEE Transactions on Information Security and Privacy*, 15;311–324.

[15]H. Park, S. Lee, and J. Kim, (2024). Real-time analytics for distributed big data systems, *ACM Transactions on Data Science*, 10(2);87–99.

[16]D. Kumar and P. Tiwari, (2024). Optimizing machine learning algorithms for big data scalability," *IEEE Transactions on Computational Intelligence*, 15(5);205–219.

[17]Z. Li, M. Sun, and Y. Zhao, (2024). Real-time processing frameworks for hybrid big data models," *Springer Journal of Real-Time Data Science*, 9;234–248.

[18]F. Luo and X. Tang, (2024). Streamlining big data analytics using hybrid methodologies, *Proceedings of the International Conference on Data Engineering*, pp. 132–141.

[19]J. Patel, R. Singh, and A. Desai, (2024). Cloud-based big data processing using hybrid techniques," *Journal of Cloud Computing and Data Management*, 18(3);102–115.

[20]N. Zhang and L. Wang, (2024). Distributed hybrid models for big data analytics, *IEEE Transactions on Parallel and Distributed Systems*, 15(6);442–453.

[21]S. Praseetha, & S. Sasipriya. (2024). Adaptive Dual-Layer Resource Allocation for Maximizing Spectral Efficiency in 5G Using Hybrid NOMA-RSMA Techniques. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.665

[22]K.S. Praveenkumar, & R. Gunasundari. (2025). Optimizing Type II Diabetes Prediction Through Hybrid Big Data Analytics and H-SMOTE Tree Methodology. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.727

[23]Vijayadeep GUMMADI, & Naga Malleswara Rao NALLAMOTHU. (2025). Optimizing 3D Brain Tumor Detection with Hybrid Mean Clustering and Ensemble Classifiers. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.719

[24]SHARMA, M., & BENIWAL, S. (2024). Feature Extraction Using Hybrid Approach of VGG19 and GLCM For Optimized Brain Tumor Classification. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.714

[25]I. Prathibha, & D. Leela Rani. (2025). Rainfall Forecasting in India Using Combined Machine Learning Approach and Soft Computing Techniques: A HYBRID MODEL. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.785

[26]Tirumanadham, N. S. K. M. K., S. Thaiyalnayaki, & V. Ganesan. (2025). Towards Smarter E-Learning: Real-Time Analytics and Machine Learning for Personalized Education. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.786

[27]Johnsymol Joy, & Mercy Paul Selvan. (2025). An efficient hybrid Deep Learning-Machine Learning method for diagnosing neurodegenerative disorders. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.701