

SCALABLE IT SOLUTIONS USING MLP NEURAL NETWORKS AND AI-BASED BIG DATA PROCESSING

Xiaobin Zhao^{1✉}, Shuhong Yuan²

Received on October 17, 2025

Presented by Ch. Roumenin, Member of BAS, on January 27, 2026

Abstract

Scalability and computational efficiency are significant challenges for modern IT systems as AI and big data applications continue to grow. This study explores the use of Multilayer Perceptrons (MLPs) as a scalable solution to enhance performance in cloud and IoT environments. The proposed hybrid MLP model, integrated with Apache Spark and TensorFlow, achieved a classification accuracy of 91.81%, surpassing K-Means (87.73%) and Decision Trees (85.77%). Advanced preprocessing techniques, including normalization, dimensionality reduction, and automated hyperparameter tuning, reduced training time by 1.5 times, thereby improving computational efficiency. The model also shows strong potential in predictive analytics tasks that require fast and accurate decision-making. Despite challenges such as interpretability and computational load, future work will focus on exploring semi-supervised learning, federated learning, and explainable AI to enhance transparency and reduce dependence on labelled data. Overall, the findings suggest that MLPs provide a solid foundation for scalable and efficient IT systems in high-dimensional, real-time data environments.

Key words: scalability, multilayer perceptrons (MLPs), artificial intelligence, big data, predictive analytics

This work was supported by Special Cultivation Project on Urban Infrastructure Lifeline, Zhejiang College of Construction; Major Humanities and Social Sciences Research Project of Zhejiang Provincial Universities (No. 2024GH066).

<https://doi.org/10.7546/CRABS.2026.02.10>

Introduction. The rapid expansion of data and the growing complexity of applications have made scalability and efficiency central challenges in the development of modern IT systems. As organizations across various sectors, including healthcare, finance, and manufacturing, increasingly rely on data-driven technologies, the demand for scalable, high-performance infrastructures continues to rise. However, traditional IT architectures often struggle with resource allocation, data throughput, and system flexibility issues that are further exacerbated by the computational demands of artificial intelligence (AI) and big data analytics [1]. AI and big data technologies offer transformative capabilities, enabling advanced analytics, predictive modelling, and intelligent decision-making. Machine learning (ML) and deep learning techniques, in particular, can uncover insights from large and complex datasets, facilitating operational optimization [2]. Meanwhile, big data frameworks support the real-time processing and storage of high-volume, high-velocity information streams. Yet, integrating these technologies into scalable IT systems introduces new performance bottlenecks that must be addressed [3]. Multilayer perceptrons (MLPs), a class of artificial neural networks, have emerged as a promising solution for enhancing both scalability and efficiency [4]. Their ability to model complex, nonlinear relationships makes them ideal for predictive analytics, anomaly detection, and adaptive system control. Furthermore, when combined with distributed platforms such as Apache Spark and TensorFlow, MLPs can support real-time data processing and intelligent resource management, even in dynamic and large-scale environments [5].

This section reviews recent advancements in scalable IT systems, with a focus on the role of multilayer perceptrons (MLPs) in addressing big data challenges. The review includes findings from key studies published after 2020, focusing on scalability, performance, and integration of MLPs with distributed computing environments.

KOTTALA et al. [6] analyzed machine learning algorithms for processing large-scale social media datasets, highlighting the trade-off between accuracy and scalability. Their results demonstrated that MLPs, especially when paired with parallel computing platforms like Apache Spark, significantly enhance performance in sentiment analysis and predictive modelling.

MENG et al. [7] investigated algorithmic optimizations for scalable AI systems. Their work emphasized techniques such as neural architecture search and model compression, noting that MLPs benefit considerably from these methods. These enhancements reduce computational overhead, making MLPs more suitable for deployment in high-volume environments.

KOTHA et al. [8] explored the use of MLPs within e-commerce and ERP systems. Their study showed that MLP-based algorithms improve operational efficiency by enabling faster predictive analytics and anomaly detection. They concluded that scalable MLP implementations lead to lower operational costs and better customer experience.

PADMANABAN [9] focused on deploying machine learning algorithms, including MLPs, on distributed infrastructures. The study highlighted how MLPs, when integrated into dynamic, resource-managed systems, maintain performance consistency and efficiently scale across varying dataset sizes.

RAMÍREZ and ISLAM [10] addressed broader challenges in AI infrastructure scalability. They underscored the value of hybrid models combining MLPs with other AI techniques to achieve balanced workloads and efficient resource allocation, especially in cloud environments.

SARMA et al. [11] introduced AI4HPC, a scalable library designed for large-scale MLP training across high-performance computing (HPC) systems. Their results showed that MLPs could scale efficiently across thousands of GPUs while maintaining high accuracy, especially when optimized with adaptive learning techniques.

BARRAK [12] examined the integration of MLP in serverless computing environments. The study concluded that serverless architecture enhances scalability by enabling parallel execution, though challenges such as communication overhead and distributed synchronization persist.

TIAN et al. [13] proposed NOSMOG, a framework combining MLPs and graph neural networks (GNNs). By using MLPs for node-level tasks, the framework achieved substantial improvements in inference speed and accuracy, particularly for structured graph data.

ALTHATI et al. [14] investigated the role of MLPs in data integration for modern big data platforms. They found that MLPs enhanced data quality, reduced processing delays, and improved system responsiveness in real-time data pipelines.

DIAMANTIS and IAKOVIDIS [15] presented ASML, a scalable machine learning architecture featuring MLPs. In cases such as obstacle detection, ASML achieved high throughput and low latency, demonstrating the viability of MLPs in large-scale, real-time systems.

Novelty and contribution. This work provides several novel contributions to the development of scalable and intelligent IT systems. Unlike existing studies that focus on isolated machine-learning models, the proposed framework integrates a multilayer MLP with multiple complementary classifiers, creating a hybrid architecture that improves robustness and adaptability in large-scale environments. The study also introduces a unified data-processing and training pipeline capable of operating on distributed Spark and TensorFlow platforms, demonstrating practical scalability for real-world big-data applications. In addition, the feature-selection strategy and the ensemble-based decision mechanism enhance predictive performance while reducing computational overhead. Together, these elements form a coherent and efficient system that advances current approaches to scalable AI-driven IT infrastructure.

Methods. The proposed classification framework is implemented using a Multilayer Perceptron (MLP) with parameters selected to support efficient con-

vergence and stable performance on large-scale datasets. The key configurations of the MLP are as follows:

- **Architecture.** Three fully connected hidden layers with 256, 128, and 64 neurons, respectively.
- **Activation functions.** ReLU for all hidden layers and Softmax for the output layer.
- **Optimizer.** Adam with an initial learning rate of 0.001.
- **Batch size.** 128 samples per iteration.
- **Early stopping.** Training terminates if validation accuracy does not improve for 10 consecutive epochs.

Model performance is evaluated using standard classification metrics, including **precision**, **recall**, **accuracy**, and **F1-score**. To assess scalability, the model is tested under increasing dataset sizes while monitoring **training time**, **memory usage**, and GPU utilization.

Hyperparameter tuning is conducted using a combination of **grid search**, **random search**, and **Bayesian optimization**, allowing the model to automatically identify optimal learning rates, neuron counts, and regularization parameters.

The entire data-processing and training pipeline is automated. The dataset is partitioned and distributed across **Apache Spark** nodes to enable parallel preprocessing. Training is performed within a **distributed TensorFlow environment**, allowing gradient computation across multiple GPUs to accelerate convergence. This methodology ensures that the proposed MLP is optimized for real-world, large-scale applications while maintaining high robustness, computational efficiency, and reproducibility. Figure 1 demonstrates how MLPs tackle big data challenges.

Description of proposed concepts. The proposed framework integrates intelligent signal processing with a hybrid classification strategy designed to improve scalability and resource efficiency in cloud-based environments. At the core of the system is a multilayer MLP classifier enhanced through interactions with additional machine learning models, including Support Vector Machines (SVMs), K-Nearest Neighbours (KNN), K-Means clustering, and Decision Trees (DT). Input signals are first transformed into high-dimensional feature vectors through preprocessing. These vectors are then forwarded to the MLP for supervised learning. In parallel, a dedicated feature extraction module – implemented using a sequence of convolution, pooling, fully connected, and dropout layers – produces refined representations that enhance downstream classifier performance. Feature relevance is quantified using F-score metrics, enabling the selection of the most

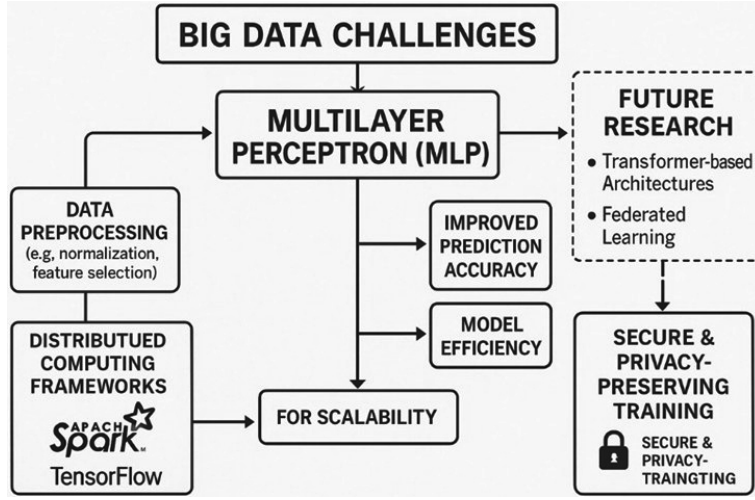


Fig. 1. Integration of MLPs for scalable and secure big data processing

discriminative attributes and reducing dimensionality. The extracted features are processed by auxiliary classifiers (SVM, K-Means, and DT), whose outputs are combined using majority voting and dynamic weighting. This ensemble strategy improves prediction robustness, supports adaptation to varying data distributions, and strengthens classification performance in high-volume environments. The hybrid design provides a flexible architecture capable of handling complex signal structures, ensuring scalability and high accuracy in distributed systems.

Mathematical model description. The transaction evaluation and security algorithm in the financial supply chain is assessed using standard classification metrics, namely Accuracy, Recall, and F1-score. In this subsection, we present the corresponding equations and define all variables involved [16].

Classification metrics. The Accuracy of the model is defined as:

$$(1) \quad \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

The Recall (also called Sensitivity or True Positive Rate) is given by:

$$(2) \quad \text{Recal} = \frac{TP}{TP + FN}.$$

The F1-score, which represents the harmonic mean of Precision and Recall, is expressed as:

$$(3) \quad F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where TP is the number of true positive samples, TN is the number of true negative samples, FP is the number of false positive samples, FN is the number of false negative samples, $\text{Precision} = \frac{TP}{TP + FP}$.

These metrics are used to evaluate the effectiveness of the proposed hybrid MLP-based model compared with baseline methods such as Decision Tree and K-Means.

Conversion of continuous outputs to class labels. In the proposed model, the hold-out method is used to divide the dataset into two disjoint subsets. Specifically, 70% of the data samples are randomly selected for training, while the remaining 30% are reserved for testing. This approach provides an unbiased evaluation of the model's generalization performance.

Since the outputs produced by the proposed architecture are continuous numerical values, they must be transformed into a discrete form appropriate for classification. To achieve this, each raw model output y_i , corresponding to the i -th sample, is mapped into a predefined range using the following relationship [17]:

$$(4) \quad L_i = \frac{(K_s - 1)}{\left(\frac{1}{\exp(m_i - 1)}\right)}$$

where m_i is the raw continuous output of the model for sample i , L_i is the normalized (mapped) output value, K_s is the upper limit of the predefined mapping range, $\exp(\cdot)$ denotes the exponential function, $i = 1, 2, \dots, N$, with N being the total number of samples.

The upper limit of the normalization range is determined by the structure of the classification problem, and m_i represents the output produced by the model prior to scaling. Because the dataset contains two classes (0 and 1), the upper limit is chosen as $K_s = 3$, ensuring that the scaled outputs fall within the interval $[0, 3]$. This predefined range guarantees a stable mapping between continuous model outputs and categorical decision values.

Following normalization, a threshold-based rule is used to convert the values of L_i into discrete class labels. Specifically:

- if $L_i < 2$, the sample is assigned to Class 1;
- If $L_i \geq 2$, the sample is assigned to Class 0.

This procedure provides a simple and interpretable mechanism for translating continuous model responses into binary classification outcomes.

Weighted sensitivity-based fitting function. To evaluate the performance of the classifiers during training, a weighted sensitivity-based fitting function is defined. This function incorporates a weighted contribution from each class, enabling the model to balance class importance or compensate for class imbalance when necessary. The fitting function is expressed as [18]:

$$(5) \quad F = \sum_{i=1}^C \left(m_i \cdot \frac{K_c(i)}{K_T(i)} \right),$$

where: C is the total number of classes (here, $C = 2$), i is the class index, $K_c(i)$ is the number of correctly classified samples belonging to class i , $K_T(i)$ is the total number of samples belonging to class i , m_i is a weighting coefficient for class i , with $m_i \in [0, 1]$, F is the overall fitting function minimized during training.

The component $K_c(i)$ represents the number of samples belonging to class i that are correctly classified, while $K_T(i)$ denotes the total number of samples in class i . The weighting factor m_i allows the model to modulate the influence of each class based on its relative significance or prevalence in the dataset.

During the training stage, the efficiency of the constructed classifiers is measured by examining the degree to which they minimize the fitting function defined above. Lower values of F correspond to higher class-wise sensitivity and, consequently, improved overall classification performance.

Results and discussion. In this phase of the study, we evaluate the performance of the proposed hybrid MLP model in comparison to conventional methods, specifically the Decision Tree and K-Means classifiers. The objective is to assess model effectiveness in terms of classification accuracy, record distribution, and clustering feature significance. The pre-processing stage effectively compresses the signals through SVM channels, facilitating immediate segmentation. Classification is performed using decision tree algorithms and feature extraction methods. We generate a transfer graph from the input and output matrices, adapting it for the final stage to produce predicted values for key factors. Our approach begins with normalizing features using mean and standard deviation functions, then selecting essential components by minimizing comparisons across the three research algorithms, which helps identify the best features while eliminating irrelevant ones. This method results in a significant reduction of approximately 1.5 times in training time. Upon completing the calculations, we develop a robust system for predicting all calculated signals (nodes), creating tables that include Precision, Recall, F1 Score, and Support metrics. Additionally, we extract data related to target analysis models' type and learning network index, as well as test network indices from multiple datasets in TXT and CSV formats. The initial classification and separation of data are grounded in triple states and preliminary evaluations of intermediate variables. The number of features utilized from the source datasets varies, with Our Method, K-Means, and D-Tree serving as dimensions for state embedding to analyze the parameters and datasets. Finally, the datasets are trained with a random distribution of 70% for training and 30% for testing, employing K-Means, Our Method, and D-Tree data for state classification. Table 1 summarizes the number of records processed by each model across the training and testing stages. It also includes classification counts per method and their respective accuracy percentages. These figures help illustrate the robustness and generalization capacity of each model under identical conditions.

Visual analytics and interpretability. To enhance interpretability and evaluate the performance of the proposed model in a visual context, a series of

T a b l e 1

Record distribution and classification accuracy across models

Model	Records processed (Train)	Records processed (Test)	Records by D-Tree	Records by K-Means	Records by proposed method	Accuracy (%)
D-Tree	5630	1878	3190	2395	3235	85.77
K-Means	6275	2093	4041	3030	3245	87.73
Proposed MLP	4275	1420	2818	2113	2144	91.81

figures was generated. These plots illustrate the optimization process, time-based accuracy, and clustering structure of the data. The data were organized into 1000 clusters and then analyzed separately for each cluster. The best node, which has the highest score among the other nodes, is displayed as a member of that cluster. Figure 2 visualizes the clustering results from 1000 randomized samples. The figure highlights the best-performing node in each cluster, confirming the ability of the proposed model to maintain high intra-cluster cohesion and inter-cluster separation.

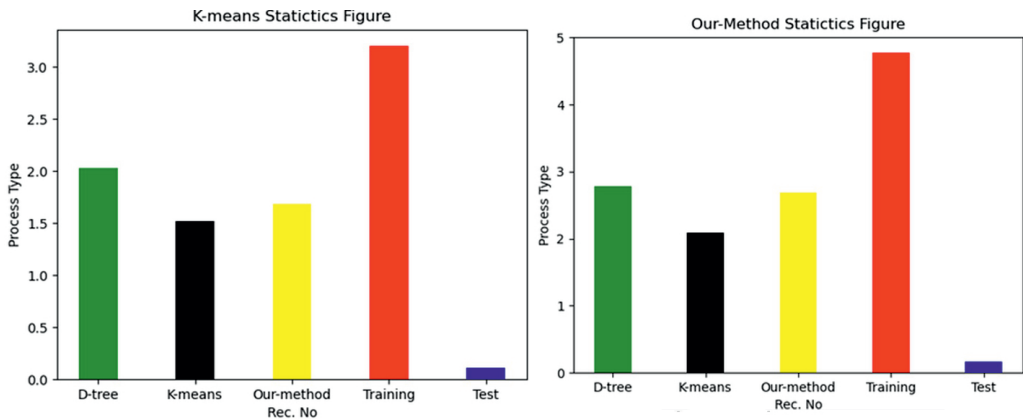


Fig. 2. K-means and our methods statistics

Conclusion. This study confirms the effectiveness of Multilayer Perceptrons (MLPs) as a robust and scalable solution for addressing the growing challenges of big data processing and modern IT system performance. The proposed MLP-based model, integrated with distributed computing platforms such as Apache Spark and TensorFlow, demonstrated strong scalability by maintaining consistent accuracy as data volume increased. Experimental results showed that the model achieved an average accuracy of 91.81%, outperforming traditional approaches such as K-Means (87.73%) and Decision Trees (85.77%). The incorporation of automated hyperparameter tuning and advanced preprocessing techniques, including normalization and dimensionality reduction, led to a 1.5 times reduction

in training time, significantly improving system efficiency. The model also proved highly effective in intelligent signal clustering and real-time resource allocation tasks, making it especially well-suited for cloud computing environments and Internet of Things (IoT) applications. The hybrid architecture, which combines MLPs with KNN and SVM layers, enhances classification and prediction precision across large-scale, high-dimensional datasets. Furthermore, the distributed framework ensured balanced resource utilization and minimized computational bottlenecks. These outcomes highlight the value of MLP-based hybrid models in building adaptive, future-ready systems that scale efficiently without compromising performance or interpretability.

REFERENCES

- [1] MENG X. (2024) Optimization of algorithmic efficiency in AI: Addressing computational complexity and scalability challenges, *Appl. Comput. Eng.*, **45**(1), 305–311.
- [2] HUDA A. M., R. SIMAMORA, K. ULWI (2024) Implementation of agent systems in big data management: Integrating artificial intelligence for data mining optimization, *J. Comput. Sci. Adv.*, **2**(1), 33–47.
- [3] SHAFIQUE M., A. MARCHISIO, R. V. W. PUTRA (2021) Towards energy-efficient and secure edge AI: A cross-layer framework ICCAD special session paper. In: *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, 1–9.
- [4] KANUNGO S. (2024) AI-driven resource management strategies for cloud computing systems, services, and applications, *World J. Adv. Eng. Technol. Sci.*, **11**(2), 559–566.
- [5] PANDEY D. (2023) Enhancing IoT data processing efficiency with AI: A study on edge computing, *Int. J. Innov. Res. Sci. Eng. Technol.*, **12**(4), 4686–4691.
- [6] KOTTALA S. Y., D. V. GOWDA, K. M. MOUNA et al. (2024) Scalability and performance evaluation of machine learning techniques in high-volume social media data analysis. In: *11th Int. Conf. on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 1–6, <https://doi.org/10.1109/ICRIT061523.2024.10522361>.
- [7] MENG L., Y. SHAO, L. YUAN et al. (2024) A survey of distributed graph algorithms on massive graphs, *ACM Comput. Surv.*, **57**(2), 1–39.
- [8] KOTHA K. R., S. C. TOKACHICHU, S. CHENNURI (2024) AI and machine learning in enhancing scalability and efficiency of integrated e-commerce and ERP systems, *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, **10**(5), 254–264.
- [9] PADMANABAN H. (2024) Machine learning algorithms scaling on large-scale data infrastructure, *J. Artif. Intell. Gen. Sci.*, **3**(1), 1–26.
- [10] RAMÍREZ J. G. C., M. M. ISLAM (2024) Navigating the terrain: Scaling challenges and opportunities in AI/ML infrastructure, *J. Artif. Intell. Gen. Sci.*, **2**(1), 209–228, <https://doi.org/https://doi.org/10.60087/jaigs.v2i1.p228>.
- [11] SARMA R., E. INANC, M. AACH, A. LINTERMANN (2024) Parallel and scalable AI in HPC systems for CFD applications and beyond, *Front. High Perform. Comput.*, **2**, 20 pp, <https://doi.org/10.3389/fhpcp.2024.1444337>.

- [12] BARRAK A. (2024) Incorporating serverless computing into P2P networks for ML training: Database tasks and their scalability implications, *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**(21), 23439–23440, <https://doi.org/10.1609/aaai.v38i21.30419>.
- [13] TIAN Y., C. ZHANG, Z. GUO et al. (2022) NOSMOG: Learning noise-robust and structure-aware MLPs on graphs, *arXiv preprint arXiv:2201.00000*.
- [14] ALTHATI C., M. TOMAR, L. SHANMUGAM (2024) Enhancing data integration and management: The role of AI and machine learning in modern data platforms, *J. Artif. Intell. Gen. Sci.*, **2**(1), 220–232.
- [15] DIAMANTIS D., D. IAKOVIDIS (2021) ASML: Algorithm-agnostic architecture for scalable machine learning, *IEEE Access*, **9**, 51970–51982, <https://doi.org/10.1109/ACCESS.2021.3069857>.
- [16] ALAKETU M. A., A. OGUNTIMILEHIN, K. A. OLATUNJI et al. (2024) Comparative analysis of intrusion detection models using big data analytics and machine learning techniques, *Int. Arab J. Inf. Technol.*, **21**(2), 326–337.
- [17] YAURI R. A., H. SURU, J. AFRIFA, H. G. MOSES (2023) A machine learning approach in predicting student's academic performance using artificial neural network, *J. Comput. Cogn. Eng.*, **3**(2), 203–212.
- [18] BANDANADAM S. R., P. K. RANGARAJAN (2025) Decentralized big data auditing scheme for cloud storage based on blockchain with adaptive EI-GAMAL and Gazelle optimization, *J. Comput. Cogn. Eng.*, Online First, <https://doi.org/10.47852/bonviewJCCE52025101>.

¹*Digital Campus Management Center, Zhejiang College of Construction, No. 151 Xuezhi Road, Xiaoshan Higher Education Park, Hangzhou City, Zhejiang Province, China*
e-mail: ybxb@163.com

²*Information Technology Center, Zhejiang University, No. 866 Yuhangtang Road, Xihu District, Hangzhou City, Zhejiang Province, China*
e-mail: 13336107509@163.com