The American Journal of Engineering and Technology
ISSN 2689-0984 | Open Access

TYPE Original Research
PAGE NO. 180-185
DOI 10.37547/tajet/Volume07Issue07-17

Application Of Ai for Enhancing the Performance of Distributed Systems

Danil Temnikov
Lead Engineer EPAM Systems Redmond, USA

Roman Dubinin
Staff Engineer, SOLAR SECURITY JSC Moscow, Russia

OPEN ACCESS

SUBMITED 15 June 2025
ACCEPTED 08 July 2025
PUBLISHED 31 July 2025
VOLUME Vol.07 Issue 07 2025

CITATION
Danil Temnikov, & Roman Dubinin. (2025). Application Of Ai for Enhancing the Performance of Distributed Systems. The American Journal of Engineering and Technology, 7(07), 180–185. https://doi.org/10.37547/tajet/Volume07Issue07-17

Abstract: This article examines how artificial-intelligence technologies can improve the efficiency of distributed computing systems that face challenges of scalability, overload and limited flexibility in responding to external changes. The aim of the study is to explore AI-based methods designed to increase performance in distributed environments. The research draws on a theoretical analysis of publications in the field of distributed computing. Machine-learning algorithms allow forthcoming load changes to be detected in advance and computing tasks to be reassigned automatically, thereby reducing response time and boosting overall performance. Employing neural networks to analyse utilisation and redistribute resources improves the operation of distributed systems and smooths peak-load periods. The findings will be of interest to professionals working with distributed computing systems, cloud technologies and to other researchers investigating methods for enhancing the reliability and performance of computing platforms. The study concludes that integrating artificial intelligence into distributed systems increases their efficiency and resilience, opening new opportunities for optimising modern computing infrastructures.

Keywords: artificial intelligence, distributed systems, machine learning, neural networks, performance optimisation, load balancing, load prediction, fault tolerance.

Introduction

Recent advances in distributed computing and cloud

technologies have led to a surge in data volumes and have made real-time information processing increasingly complex. Despite their high degree of parallelism and scalability, these platforms still face challenges such as uneven load distribution, limited adaptability to changes in computing resources and sensitivity to peak workloads. Fluctuating external factors—ranging from shifts in demand to node failures—necessitate new methods for managing and optimising system performance. Employing artificial intelligence to enhance distributed systems has therefore become a key focus of contemporary research, alongside studies on resource optimisation, security and the adoption of emerging computational technologies. These areas are examined below.

Uzgoren M. et al. describe how AI automates resource management, minimises downtime and improves utilisation of computing power by proposing algorithms that efficiently redistribute tasks across system nodes [1].

Wang Y., He S., Wang Y. devote attention to AI-driven dynamic data management, investigating ways to cut energy consumption and improve resource scheduling [2].

Rasmus M., Kopertowski Z., Kozdrowski S. explore AI for routing and traffic control in programmable networks; deep-learning methods predict network congestion and reroute data flows [6]. Zhang C., Dong M., Ota K. propose Q-learning to improve computation synchronisation in distributed systems [7].

Dhaya R., Kanthavel R. show how AI analyses load distribution among providers and redirects tasks to boost system performance, thereby aiding load balancing and reducing latency [3].

Baccour E. et al. study distributed computing in the context of the Internet of Things, demonstrating how AI minimises resource use while enhancing data processing on resource-constrained devices. Distributed algorithms optimise data transfer between IoT nodes, cutting traffic volume and energy costs [4].

Wei W., Liu L. outline current issues in information security and access management for distributed systems, discussing differential-privacy mechanisms and decentralised access control that defend data against external threats [8].

AĞCA M. A. detail trusted-system architectures, authentication mechanisms and attack resilience, systematising approaches aimed at improving the reliability of distributed systems. They highlight limited scalability in existing solutions and difficulties in applying them to heterogeneous environments [9].

Bidollahkhani M. and Kunkel J. M. [5] describe how data-analysis models can predict equipment failures, facilitating maintenance planning and reducing their impact on system operation.

Sultan M. and Sultan M. [10] examine how quantum computing can process large data volumes, a capability that is vital for real-time optimisation tasks. Neuromorphic processors enable energy-efficient computation, making them suitable for IoT devices and other resource-constrained systems.

Wang Y., He S. and Wang Y. [11] investigate the potential of using Storm to boost the performance of distributed systems; data processing represents a significant share of operating costs. Dehghani M. and Yazdanparast Z. [12] outline how AI can address complex problems through advances in hardware acceleration and machine-learning algorithms, achieved by dividing algorithms into classification and clustering groups.

The literature review therefore demonstrates a wide range of AI approaches for improving the performance of distributed systems. Differences in emphasis are evident: some authors focus on resource-management automation, failure prediction and load balancing, whereas others highlight energy efficiency, optimisation of operating expenses and the security of distributed platforms.

The study's scientific novelty lies in applying machine-learning methods to optimise distributed systems in real time.

The purpose of the work is to explore how AI techniques can enhance performance in distributed environments.

Its practical significance involves applying the proposed methods to increase the efficiency of cloud platforms, server infrastructures and other distributed computing systems that require high adaptability and effectiveness under changing conditions.

The research novelty consists in employing machine-learning techniques for the dynamic improvement of distributed-computing systems.

The author's hypothesis is that implementing machine-learning algorithms in distributed systems will raise their performance by optimally allocating computing

resources and predicting peak loads.

The methodology is based on a comparative analysis of scientific publications by other researchers.

Research Results

The application of artificial intelligence to improve the performance of distributed systems is a key area in contemporary technology development. Machine-learning methods are increasingly embedded in distributed-system architectures, solving numerous tasks while enhancing efficiency, resilience and scalability. Within machine learning, *classification* is the process of predicting categories for new objects on the basis of labelled data. A model must therefore be able to make accurate predictions on data not seen during training. A defining feature of classification is its categorical nature: results always belong to discrete groups with no intermediate values [1, 2]. Figure 1 presents a taxonomy of these algorithms.

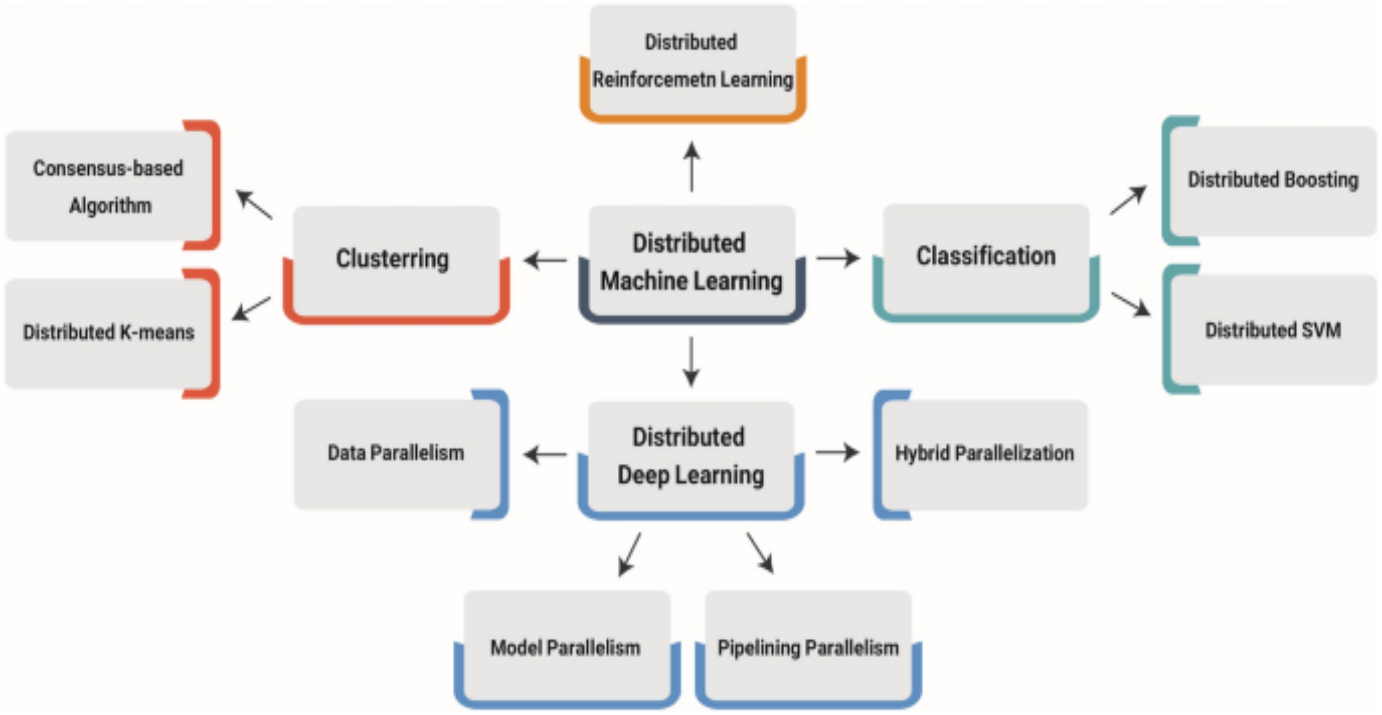

Fig. 1. Distributed machine-learning algorithms [12].

Boosting combines several weak classifiers to form a more powerful model. The central idea is that merging simple classifiers can yield better performance than deploying a single strong one. The AdaBoost method has been adapted for distributed computing: classifiers created on individual network nodes are assembled into an ensemble, improving overall accuracy.

The *support-vector machine* (SVM) is used for binary classification. It constructs a hyperplane that separates the classes in the training data and solves the problem via convex quadratic optimisation, thereby avoiding the local minima that arise in other techniques. Support vectors—points closest to the separating hyperplane—play a crucial role. To accelerate SVM on large datasets, distributed approaches have been proposed. One example is **DPSVM**, which performs efficiently in distributed environments with limited inter-node communication, minimising data-exchange costs [3, 6].

Consensus clustering merges several clusterings to form a single, more stable solution. The approach improves clustering results by running the algorithm iteratively on subsets of the original data. The **k-means** algorithm is widely used because of its efficiency and simplicity: it computes centroids, assigns each data point to the nearest centroid and then recalculates centroids as the mean of all points in each cluster.

Deep neural networks (DNNs) are multilayer structures that echo brain function: neurons process inputs and pass signals through weighted connections. Effective training, however, demands significant computational power, necessitating various forms of parallelism. Optimisation techniques for DNN training include data parallelism, model parallelism, pipeline parallelism and hybrid variants. Partitioning data into mini-batches accelerates training by distributing computation across multiple nodes and reducing system load [11].

Model parallelism is a method used to accelerate DNN training by splitting the model across multiple nodes

(Fig. 2). The approach focuses on partitioning the network into logically connected subsystems, taking into account both structural and functional architectural features. A central task is to devise efficient algorithms for distributing these subsystems so that the training load is balanced evenly, thereby raising overall efficiency. It should be noted, however, that model-level parallelism is subject to fundamental scalability limits imposed by high inter-device communication latency, which ultimately degrades system performance.

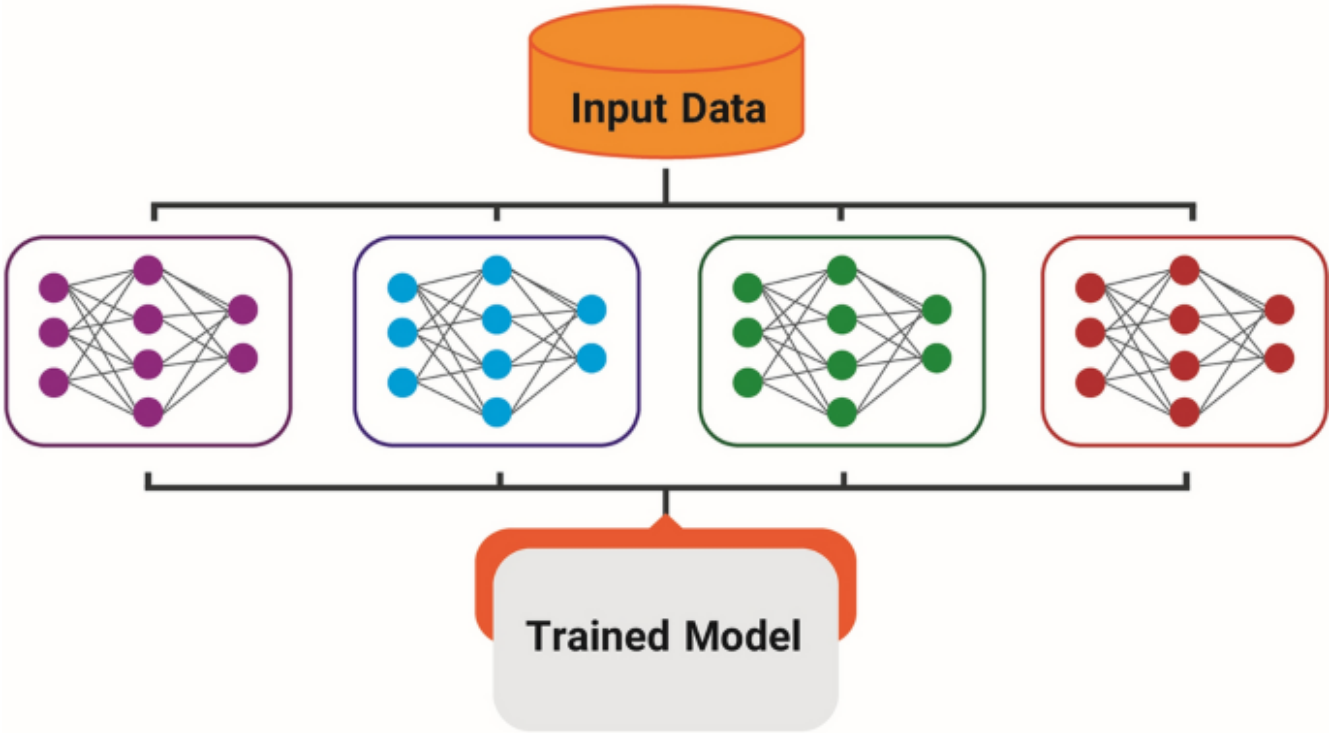

Fig. 2. Model parallelism scheme [12].

Pipeline parallelism divides the neural-network training workflow into several stages and passes intermediate results between them. This increases throughput during parallel training by optimising the overlap between computation and data transfer, thus reducing communication overhead. Experiments show [12] that the PipeDream method delivers 5.3 times higher performance than traditional in-batch parallelism; on a four-GPU platform it sped up training by 8.91 times relative to data parallelism.

Hybrid parallelisation combines data and model parallelism to cut communication costs when training deep neural networks. Using four GPUs accelerated training by 4.13–4.20 times compared with a single GPU [12].

Strategies such as Parallel S-SGD and BSP improve training efficiency by optimising workload distribution and minimising communication delays between nodes. The A3C algorithm, in particular, enables multiple agents to interact with the environment in parallel; each operates in its own state space, and asynchronous updates to a shared global model reduce overfitting risk while boosting overall computational performance.

Building on A3C, the IMPALA algorithm achieves high efficiency by strategically distributing parameters among learning agents. Tests on the DMLab-30 benchmark show that IMPALA outperforms A3C on key performance metrics, underscoring the advantages of this distributed-learning approach.

The DPPO and Ape-X methods apply distributed reinforcement-learning techniques, optimising data collection and gradient calculation through coordination among many worker nodes. This architecture makes effective use of shared memory for data storage, which is crucial for enhancing both the quality and robustness of learning algorithms.

Acme provides an environment that streamlines the development of distributed reinforcement-learning algorithms, emphasising readable, reproducible code [12].

Combining these approaches with quantum computing can further amplify AI capabilities, enabling faster and more efficient solutions to problems that demand substantial computational power [7, 8].

Below is Table 1, which presents the possibilities of using AI in improving the performance of distributed systems.

Table 1. The possibilities of using AI to improve the performance of distributed systems [4, 5, 9, 10]

Aspect	Machine-learning models	Intelligent algorithms for load distribution	Flexibility and adaptability	Reaction to failures and recovery	Resource-use efficiency
Automatic scaling	Use of algorithms for analysing load and scaling	AI-driven dynamic resource redistribution	Infrastructure changes according to demand	Automatic addition of new nodes under load	Automatic increase or decrease in the number of compute nodes
Load prediction	Forecasting peak workloads	Automation of redistribution depending on load	Real-time response to changing conditions	Autonomous scaling in case of failure	On-demand use of computing power
Network optimisation / Traffic management	Optimising network bandwidth	Determining optimal data-transfer paths	Rapid parameter changes to optimise speed	Fast restoration of routes after network failures	Optimising use of network channels
Resource management	Allocation of compute resources based on system data	Autonomous redistribution of resources to improve performance	Instant power reallocation in response to needs	Intelligent redistribution when nodes fail	Dynamic resource allocation according to efficiency
Performance analysis / Anomaly detection	ML-based analysis of logs and system performance	Improved server-load forecasts	Real-time intelligent tuning	Quick diagnosis and system restoration	Identification of inefficient processes for optimisation
Failure management	Forecasting faults and outages	Minimising downtime through failure prediction	Predicting failures from historical data	Forecasting and managing recovery time	Use of AI for post-failure resource restoration

In summary, introducing artificial intelligence into distributed systems opens new pathways for tackling complex operational tasks. Successful integration will require overcoming challenges such as better interpretability, energy-efficient operation and strong security guarantees. Continued advances in AI promise increasingly autonomous computing platforms with high adaptability, broadening horizons for many sectors and research domains.

Conclusion

The analysis shows that integrating AI technologies into distributed computing systems positively affects performance, flexibility and self-healing capability.

Machine-learning algorithms, neural networks and reinforcement-learning methods influence how computing power is distributed and deliver effective real-time workload forecasting and control. Dynamic load management clearly outperforms static methods. Implementing AI algorithms improves performance and increases system fault tolerance, enabling rapid responses to changing operating conditions and predicting failures before they occur. The results confirm that artificial intelligence enhances the efficiency and adaptability of distributed systems amid evolving computing environments.

References

1. Uzgoren M. et al. Examination of AI Enhanced Distributed Systems and its Effects on Software Engineering //Proceedings of London International Conferences. – 2024. – Vol. 11. – pp. 109-119.
2. Wang Y., He S., Wang Y. AI-Assisted Dynamic Modelling for Data Management in a Distributed System //International Journal of Information Systems and Supply Chain Management (IJISSCM). – 2022. – Vol. 15 (4). – pp. 1-18.
3. Dhaya R., kanthavel R. AI - Based Framework for Private Cloud Computing. - 2021. - pp.1-25.
4. Baccour E. et al. Pervasive AI for IoT applications: A survey on resource-efficient distributed artificial intelligence //IEEE Communications Surveys & Tutorials. – 2022. – Vol. 24 (4). – pp. 2366-2418.
5. Bidollahkhani M., Kunkel J. M. Revolutionizing System Reliability: The Role of AI in Predictive Maintenance Strategies //arXiv preprint arXiv:2404.13454. – 2024. - pp.1-9.
6. Rasmus M., Kopertowski Z., Kozdrowski S. Ai application in next generation programmable networks //2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). – IEEE, 2022. – pp. 1-3.
7. Zhang C., Dong M., Ota K. Employ AI to improve AI services: Q-learning based holistic traffic control for distributed co-inference in deep learning //IEEE Transactions on Services Computing. – 2021. – Vol. 15 (2). – pp. 627-639.
8. Wei W., Liu L. Trustworthy distributed ai systems: Robustness, privacy, and governance //ACM Computing Surveys. – 2024. - pp. .1-39.
9. AĞCA M. A. Trusted distributed artificial intelligence for critical and autonomous systems. – 2023. - pp. 1-7.
10. Sultan, M., & Sultan, M. (2024). Advanced Computation Techniques for Complex AI Algorithms // International Journal of Science and Research (IJSR). - 2024. - pp.1-6.
11. Wang Y., He S., Wang Y. AI-Assisted Dynamic Modelling for Data Management in a Distributed System //International Journal of Information Systems and Supply Chain Management (IJISSCM). – 2022. – Vol. 15 (4). – pp. 1-18.
12. Dehghani M., Yazdanparast Z. From distributed machine to distributed deep learning: a comprehensive survey //Journal of Big Data. – 2023. – Vol. 10 (1). – pp. 158.