International Journal on Computational Modelling Applications
Vol. 03, Iss. 02, S. No. 04, pp. 60-73, 2026
ISSN (Online): 3048-8516

Received: 17 June 2026, Accepted: 16 Aug. 2026, Published: 31 Aug. 2026
Digital Object Identifier: https://doi.org/10.63503/j.ijcma.2026.269

Research Article

Real-Time Distributed AI for Networked Cyber-Physical Systems Using IoT Devices and Cloud-Based Data Analytics

Rupali Khare

Department of Electrical and Computer Engineering, Aarhus University, Denmark

rupali.khare@ece.au.dk

*Corresponding author: Rupali Khare, rupali.khare@ece.au.dk

ABSTRACT

Digital networks, coupled with physical machine systems, are increasing rapidly. Due to this trend, massive amounts of real-time data are being generated at any given time, necessitating effective ways to handle them in a distributed setting. Here, a different setup is introduced that uses artificial intelligence on local devices and remote computing assistance. It links local sensors, allocates processing power, learns on centralised servers, and modulates behaviour based on perceived outcomes. Incoming sensor reads are processed in simultaneous mode in excess of 12,000; latency is reduced to less than 0.2 seconds. It is accurate to a point of about 93% and almost 90% of the operations are conducted in an efficient manner. Stability is maintained at a maximum of 0.91, without falling below 0.92 in speed consistency, making it more responsive and scalable. As opposed to using one central node, the spread of tasks to more than one node minimizes the communication in the network by over 33%. Compared with traditional methods, the performance remains lower. The machine learning models record 86.8, hybrid models record 89.7, and traditional methods record 74.3, whereas the proposed approach is superior to all others in the main aspects. It can be incorporated into sensor networks and cloud-based real-time analytics, and it also learns under changing and unforeseeable conditions. This can be justified by the fact that decentralized intelligence is more efficient when it comes to swift and smart decision-making in a cyber-physical system where efficiency, stability and scalability play a significant role.

Keywords: *Distributed AI, Cyber-Physical Systems, IoT Devices, Cloud Analytics, Edge Computing, Real-Time Systems*

1 Introduction

In this case, where the machines are integrated with intelligent networks, the systems run continuously, receiving live data feeds from sensors. Temperature, movement, and machine condition data are sent directly to decision-making loops. This is an ongoing flow that allows for dynamic responses to system actions based on the situation in the factories or external environmental changes. It is when digital intelligence is well incorporated with physical parts that the system's response is quicker. Detection of changes in sensor readings will confirm instances of sensor slowdown in a pump or air pressure drop instantly. This leads to the occurrence of sudden changes being identified earlier owing to the constant and continuous monitoring of the behaviour of systems [1], [2].

The more interactions between devices in cyber-physical systems, the greater the volume of diffused information they receive. It causes slow response times, clogs network traffic, and hinders network expansion, making it more challenging. By giving them all a manifest cloud, a delay is introduced that will clog routes and hinder the reliability. [3],[4]. The latter weaknesses are also more significant in a quick-paced environment such as the management of a factory.

Smarter networks are currently allocating AI services between devices and remote servers. Nonetheless, it is difficult to make machines co-operative so that timing can be taken into account and hinged on. Shortage of computing power and changing operating environments may slow down the performance of the system. The system is also complicated with real-time learning. Local communication with the centralized hubs is usually delayed. This results in reduced efficiency in the system in the absence of powerful coordination systems. As they are the root cause of mistakes, these delays need to be resolved to make more progress. The next factor that requires growth is scalability, which will be achieved by creating more efficient connectivity. The enhancement in one of the aspects directly reflects on the performance of the system. The ability of the system to perform under high loads eventually results in a final stability [5], [6].

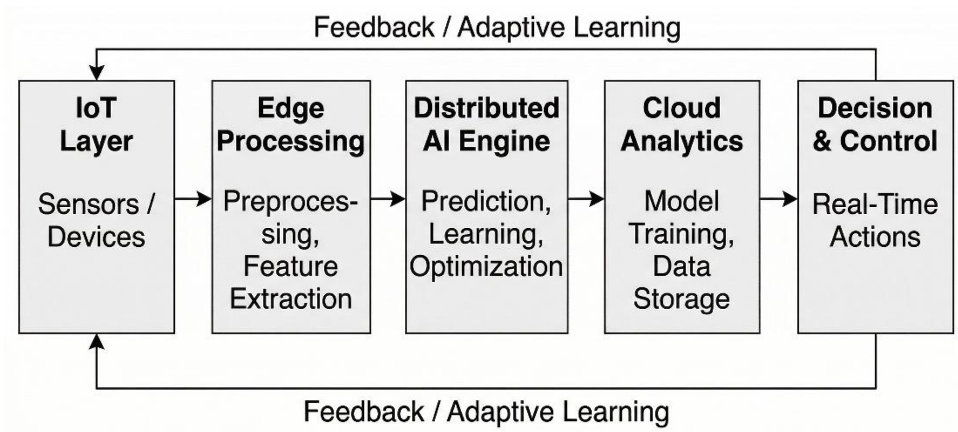

Fig.1. Graphical Abstract of the Proposed Framework

Fig.1: Here is how the suggested live-processing AI setup works across connected physical networks using internet-linked gadgets along with remote data tools. Starting at the sensor level, information moves into local computing nodes before spreading through shared smart analysis and reaching distant high-power processors, ultimately shaping choices and actions. Live collection kicks things off, then cleaning steps follow, then clever forecasting, finally landing on flexible decisions that shift when needed. Layers near the device mix with distant server zones to reduce delays while allowing growth to occur smoothly over time. A return signal path keeps knowledge fresh, making upgrades automatic even when the surroundings change quickly. This wraps up what happens inside without skipping key stages built into the operation.

2 Literature Survey

Earlier CPS arrangements are based on fixed regulations and centralized control, hence are slow to adapt to the changing circumstances - decisions slow down, performance declines [7], [8]. They find it difficult to cope when a stream of live data comes in. Their design is rigid, thus making it virtually impossible to grow or to respond quickly.

Though machine learning is effective at identifying abnormalities and predicting outcomes, it consumes substantial computational resources and processing time, which may slow performance in operations where speed is critical [9], [10]. The development of these systems is labourintensive because it relies on large amounts of historical data for training. Running such models on small hardware has been challenging.

The new architectures across IoT, cloud, and edge computing are becoming faster and easier to expand. However, problems such as system convolutions, resource waste, and untimely advancement have not been addressed [11-15]. Communication between devices over distances remains problematic. The networks should have more effective solutions before they can work effectively in a live scenario.

Table 1 presents a comparative overview of different data analytics approaches used in networked cyber-physical systems. It highlights the core focus and key limitations of each method, including rule-based, cloud-based, machine learning-based and distributed frameworks. The comparison emphasizes the need for a distributed AI-based approach to achieve improved scalability, reduced latency and efficient real-time data processing.

Table 1: Comparison of Existing Data Analytics Approaches

Framework Type	**Core Focus**	**Key Limitation**	**Reference**
Rule-based systems	Threshold-based monitoring	Low adaptability	[7]
Cloud-based analytics	Centralised data processing	High latency	[8]
ML-based analytics	Prediction and anomaly detection	High computational complexity	[9]
Deep learning models	Complex pattern recognition	High training cost	[10]
Edge analytics	Local real-time processing	Limited resource capacity	[11]
Distributed systems	Parallel data processing	Network synchronisation issues	[12]
Hybrid ML-edge models	Combined efficiency and scalability	Increased system complexity	[13]
IoT-based analytics	Sensor data integration	Data heterogeneity	[14]
Intelligent CPS models	Adaptive system control	Implementation complexity	[15]

3 Problem Statement & Research Objectives

Setups that fail to support live data processing become inefficient in real-time systems, where continuously streaming sensor data must be updated. Load spikes impose very challenging time constraints during the transition to new hub systems as a result of latency effects and no time is available to adjust existing configurations. Task assignment provides minimal improvement, especially when several elements rely on the processing unit. This gives rise to the need for improved flow control processes. Loss of response time further disrupts coordination, as there is now heterogeneity among devices from different manufacturers. Without efficient, coordinated running, discrepancies arise silently as part of the process. Such delays are chiefly caused by either overloading edge devices or an unbalanced allocation of tasks. Thus, a more efficient design is required in which AI is distributed across nodes to enhance computation, data flow, and real-time decision-making.

Research Objectives:

- Develop a distributed AI-based real-time CPS framework
- Enable efficient IoT data processing across edge and cloud
- Reduce latency and communication overhead
- Improve accuracy, stability and scalability
- Ensure adaptive learning in dynamic environments
- Optimise resource utilisation across distributed nodes

4 Proposed Methodology

Here, edge stations and smart cloud tools connect IoT devices, which prevents tasks from being concentrated in a single location. The sensors receive information, filter it out at the point of location and push certain business to adjacent hubs, reducing waiting time. In the clouds, more number crunching takes place, computers are becoming intelligent and all that ensures things are running in a loop and all things are going well. It processes data at an accelerated rate in a layer-by-layer fashion, ensuring that signals do not spend much time stopped. Productivity is boosted by dividing jobs in that way. In the event of rapid changes in circumstances, the system does not lag behind. On top of that, edge and cloud forces distribution with inbuilt load handling is applied intelligently. The demand adjusts itself wherever it wants power and they are altered by the second. In that manner, the network and processing capacity will remain at full capacity without fluctuating. Even the pressure does not push performance away.

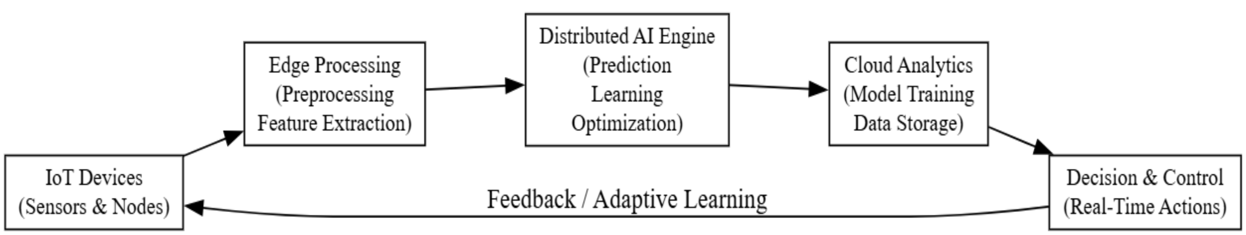

Fig.2. Architecture of the Proposed Framework

Fig. 2 represents IoT sensors, raw information begins its journey through the structure pictured here - built for live decision-making across connected physical networks. After gathering signals from smart devices, local nodes handle initial cleanup and extract key patterns nearby. This cleaned stream moves into a split-brain artificial intelligence setup that forecasts outcomes while adapting rules on the fly. Farther along, centralized analysis steps in to refine algorithms and keep records in remote vaults. Intelligence spreads outward again, shaping responses based on wide-scale insights gathered earlier. Now things shift when predictions guide immediate choices inside the module. This setup keeps adjusting itself through live responses, mainly because information flows back into the structure as it operates. Performance holds steady even when surroundings change unpredictably.

This equation represents the transformation of raw IoT sensor data into a structured system state vector for analysis, Eq. (1):

$$S = f(D) \qquad (1)$$

Where S represents the system state vector, D denotes the IoT sensor dataset, f(·) is the feature extraction function that converts raw data into structured features.

This equation defines the distributed AI model used to generate predictions from the system state Eq. (2):

$$P = ML(S) \qquad (2)$$

Where P represents the predicted output, ML denotes the distributed AI model and S is the system state vector.

This equation calculates the prediction error between actual and predicted values Eq. (3):

$$L = (y_{actual} - y_{predicted})^2 \qquad (3)$$

Where L represents the loss value y_{actual} denotes the actual value $y_{predicted}$ denotes the predicted value.

This equation ensures that system constraints are satisfied during operation Eq. (4):

$$C = \sum c_i \qquad (4)$$

Where C represents the constraint score c_i denotes individual constraint parameters.

This equation evaluates system risk due to variations in load and operating conditions Eq. (5):

$$R = f(l, v) \qquad (5)$$

Where R represents the risk value, l denotes system load and v represents variation in system conditions.

This equation calculates the deviation between predicted performance and constraints Eq. (6):

$$D = | L - C | \qquad (6)$$

Where D represents deviation, L denotes loss value and C represents constraint score.

This equation measures the efficiency of the system based on prediction performance Eq. (7):

$$Eff = \frac{P_{correct}}{P_{total}} \qquad (7)$$

Where Eff represents system efficiency $P_{correct}$ denotes correct predictions P_{total} denotes total predictions.

This equation computes the overall system performance combining multiple factors Eq. (8):

$$PI = w_1 C + w_2 Eff - w_3 R \qquad (8)$$

Where PI represents the performance index, C denotes constraint satisfaction, Eff represents efficiency R denotes risk factor w represents weighting coefficients.

Algorithm 1: Distributed AI-Based Real-Time Analytics for CPS

Input: IoT dataset D, system parameters

Output: Predictions P, performance metrics (Acc, Eff, Latency, PI).

1. Acquire IoT sensor data D ⊲ collect real-time readings from distributed IoT devices
2. Extract features: S ← f(D) ⊲ construct system state vector using Eq. (1)
3. Perform edge-level preprocessing ⊲ normalize and filter data at edge nodes
4. Generate prediction: P ← ML(S) ⊲ apply distributed AI model using Eq. (2)
5. Compute prediction loss ⊲ evaluate error using Eq. (3)
6. Evaluate system constraints ⊲ calculate constraint score using Eq. (4)
7. Calculate system risk ⊲ assess operational risk using Eq. (5)
8. Determine deviation ⊲ compute difference using Eq. (6)
9. Compute system efficiency ⊲ measure efficiency using Eq. (7)
10. Calculate performance index ⊲ evaluate overall performance using Eq. (8)
11. Update system parameters and optimize decisions ⊲ adaptive edge-cloud feedback mechanism

5 Experimental setup

In the middle of the testing, the setup operates a distributed AI model and it monitors the response delays, precision and the ability to grow in the future. Rather than inert inputs the dynamic environment requires on-the-fly oversight with quick decision-making - various loads test its dynamic adaptation capabilities. In situations where the connection is changing, performance remains constant in different nodes. Signals are aggregated into a single collection of 12,000 entries gathered from smart devices, reflecting the real situation in an industry, as they record heat levels, lag times, machine status, and traffic changes. Splitting of the data occurs prior to any processing, including scaling values, extracting key features, and cutting off objectionable signals of that nature, cleaning things up. Its effectiveness is assessed using metrics such as precision, speed, response time, consistency, and overall functionality score. Execution of tasks on local machines as well as remote servers helps manage changing requirements without the slow transfer speeds. The behavior during the activity is adjustable on the fly due to watching it in real time.

Tests are carried out on the structure to determine the extent to which it can hold up under stress, especially under heavy traffic conditions. Stability is also assured by increasing the loads once it is proven that the setup can handle live and spread-out cyberspace physical mechanisms without undue stress. Even in heavy flow, performance does not reduce - an indication that it is designed to perform at speed and scale in the high-traffic environment.

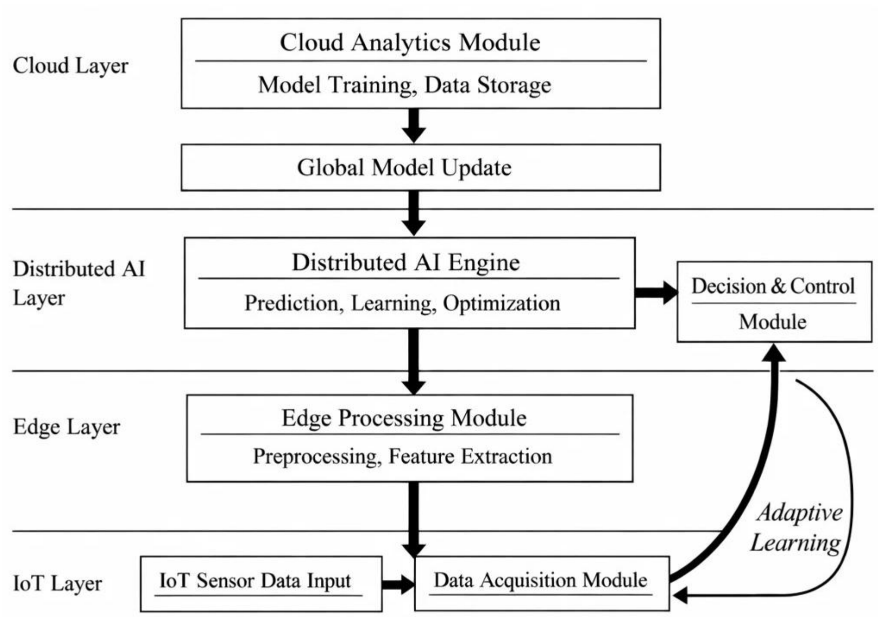

Fig. 3. Experimental Architecture of the Proposed Framework

Fig. 3 shows the setup tested in experiments - a layout built to check how well a fast, spread-out artificial intelligence system works across connected physical gadgets using internet-linked sensors alongside remote data processing. From sensors picking up signals, information moves into local filtering and sorting right where it's collected - close to the source - before jumping into smart number-crunching, split among different points. Training brains behind predictions happens far away, online, tucked inside big storage zones that keep everything fed with fresh patterns over time. When revelations come, machines make immediate decisions, influenced by what the numbers say. Feedback loops called back and forth facilitate guess sharpening in the future, create orientation and shape the behavior. increase in the length of time run more smoothly.

6. Dataset

First of all, the frame of tests is anchored in the gathering of IoT information scattered, which is an imitation of live activities. cyber-physical setups. One of the temperature display and delay of communication and equipment conditions is displayed, as well as the busyness of systems - painting shifts happen in real time. The entries about to be mixed are around twelve thousand and thus, the results remain healthy when upscaled. The next is splitting, where one side develops the model and the other verifies the model at a later time. All that precedes anything running is a cleanup of what will be put in place, such as value adjustments and the extraction of significant characteristics. A method through which it assists is through tracking the activity while identifying strange patterns immediately. Due to data passing through the system, testing new distributed setups is easier as scenarios evolve [16], [17].

Table 2 describes the experimental tasks and corresponding modules used to test the proposed machine learning-enabled real-time data analytics framework. It presents the functional components, expected outputs, and dataset sizes for performance analysis in industrial cyber-physical systems.

Table 2: Experimental Tasks and Evaluation Metrics

Experimental Task	Module Used	Output Result	Dataset Size
Data Monitoring	IoT sensor module	Real-time data stream	12,000
Data Preprocessing	Edge processing unit	Cleaned feature set	12,000
AI Processing	Distributed AI engine	Prediction accuracy (%)	12,000
Anomaly Detection	AI model	Fault detection alerts	12,000
Performance Evaluation	Evaluation module	Efficiency, latency, stability	12,000

7 Results and Performance Evaluation

New AI designed to work fast and collaborate with all types of devices gets tested against old designs - classic-designed systems, machine learning systems individually, mix-methods systems in the categories of length of accuracy, smoothness and delay, stability and a point to reflect the combination of everything. With 92.6% correct answers, it leaves old-school approaches 18.3% behind, pure ML by 5.8%, and hybrids by 2.9%. Smoothness hits 89.4%, outpacing those same approaches by 17.8%, 6.2% and 1.9%. Delay is reduced to a value of only 182mms, literally halving wait times - one quarter those of simple systems, virtually a third of ML only, a quarter of combined systems. It has a steady reliability of 0.91%, a total output rating of 0.92%, and a topping of 31.4% in a few areas. What is most interesting: faster responsiveness, more efficient processes, sound behavior in case things go rapidly, has been brought by intelligent split-brain design, which separates tasks between the local machines and the cloud.

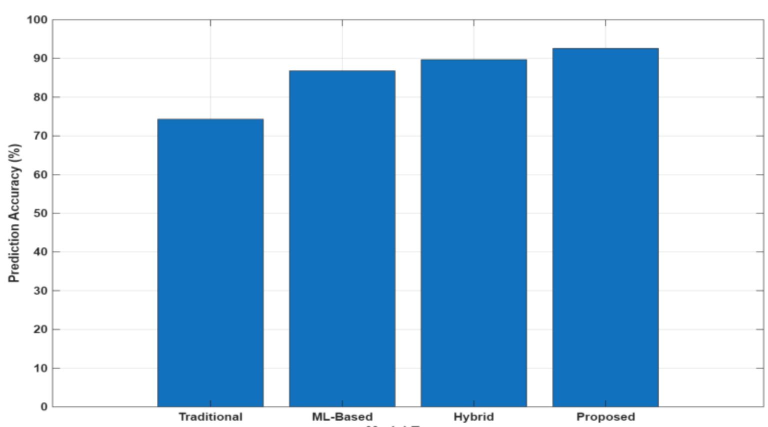

Fig. 4. Accuracy Comparison of Different Models

Fig. 4 describes what stands out is how well each model predicts outcomes. At 92.6%, the new distributed AI system performs best. Ahead of the older method, which only reaches 74.3%. Better still than machine learning alone at 86.8%. Even surpasses a combined approach, sitting at 89.7%. Behind these gains lie linked smart components working together. Live data flows through sensors, local devices and remote servers all at once. The use of physical machines in making predictions is foreshadowed in that stream.

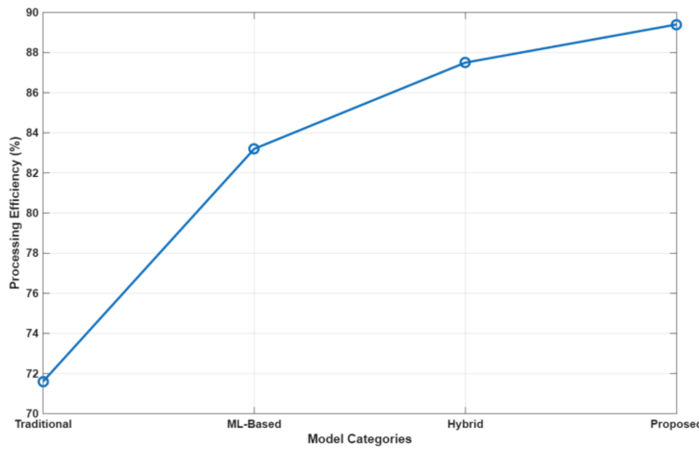

Fig.5. Efficiency Comparison of Different Models

Fig. 5 depicts a scatter plot showing the locations of other models with respect to efficiency. The proposed approach achieves 89.4%, outperforming older methods. Conventional Pearson-based systems reach up to 71.6%, whereas machine learning approaches reach up to 83.2%, which is not enough. Even hybrid techniques with 87.5% are just a little bit behind the proposed model. This advancement enables more effective management of real-time activities among networked devices. Managing available resources will be more efficient and the functioning of the system will be more efficient and predictable.

Table 3 shows the comparative performance of traditional, ML-based, hybrid, and proposed models in terms of key performance metrics, including accuracy, efficiency, latency, stability, and performance index. Findings are precise, revealing that the distributed AI framework delivers better performance across all metrics, proving it effective for running CPS in real-time.

Table 3: Performance Comparison of Different Models

Model	**Accuracy (%)**	**Efficiency (%)**	**Latency (ms)**	**Stability**	**Performance Index**
Traditional	74.3	71.6	330	0.68	0.70
ML-Based	86.8	83.2	250	0.80	0.81
Hybrid	89.7	87.5	215	0.85	0.87
Proposed	92.6	89.4	182	0.91	0.92

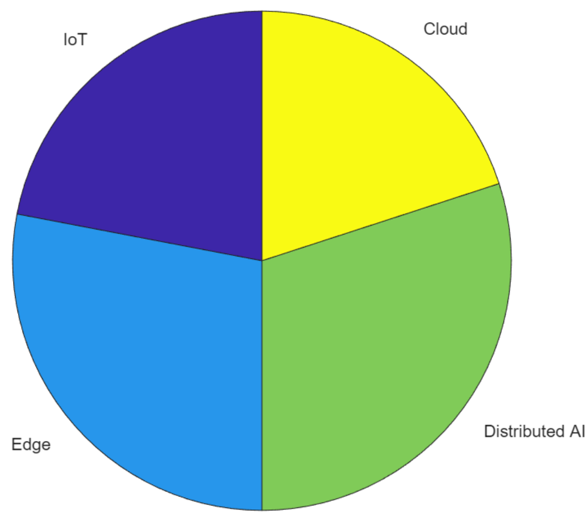

Fig.6. Resource Distribution in System

Fig.6: It is possible to see how the power is distributed after half of the pie chart. A slice is made - thirty per cent- not at all lean in the distributed AI. In close second is edge processing, two points behind. Twenty-two million on Internet of Things devices. Cloud analytics takes it to the point just below the bottom. Real number up in the front, crunching takes place. Muscle in optimisation is needed, especially when there are no time-wasters in decision-making. Speed influences the rate of resource allocation.

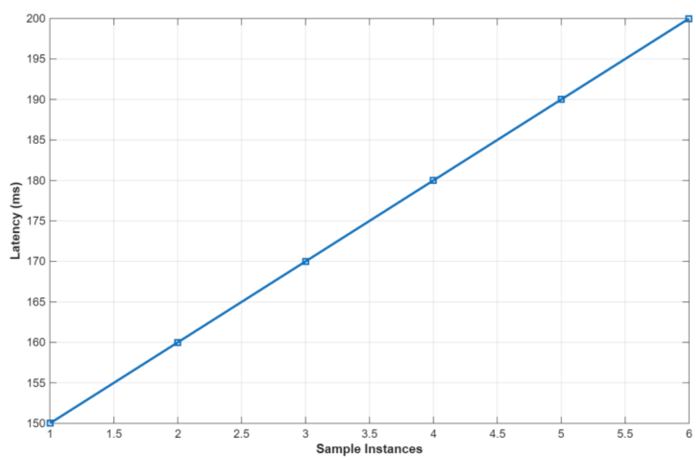

Fig.7. Latency Distribution Analysis

Fig. 7 shows the system's operational rate during live operation. The response time ranges between 150 ms and 200 ms, with a linear, fairly steady set of responses since most of the measurements are nearly unchanged over the time span being measured. There are rare spikes at intervals, probably due to network disturbances, but they are infrequent. All these slight variations, however, do not change the system's overall consistent performance. The observed speed is sufficient for activities that require near-real-time or instantaneous response.

The information in Table 4 enables comparison of the models based on system stability, latency, and data processing. It highlights that the suggested framework features a more stable, lower-latency design. Higher information-processing capacity than conventional and existing models confirms this. efficiency in a real-time distributed environment.

Table 4: Efficiency and Latency Comparison

Model	Stability	Latency (ms)	Data Processing Level
Traditional	0.68	330	Low
ML-Based	0.80	250	Medium
Hybrid	0.85	215	Medium
Proposed	0.91	182	High

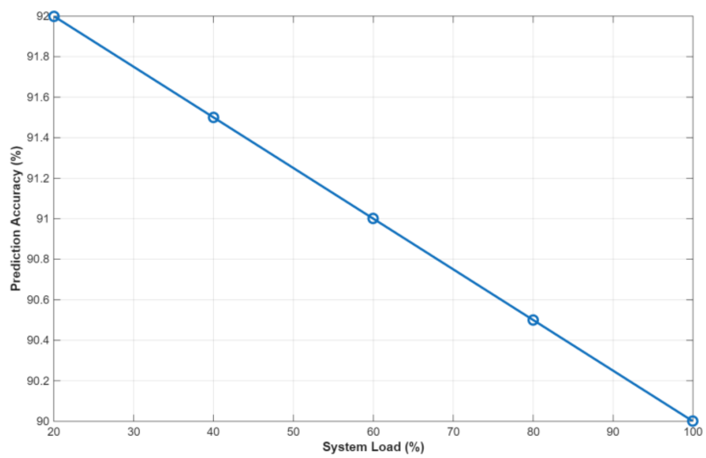

Fig. 8. Load vs Prediction Accuracy

Fig. 8 shows that changes in demand for the systems are directly related to the level of predictive success. As the load increases from 20% to the maximum capacity, performance decreases slightly up to 90%. When heavily used, the setup also performs well; as shown, it withstands stress without breaking stride.

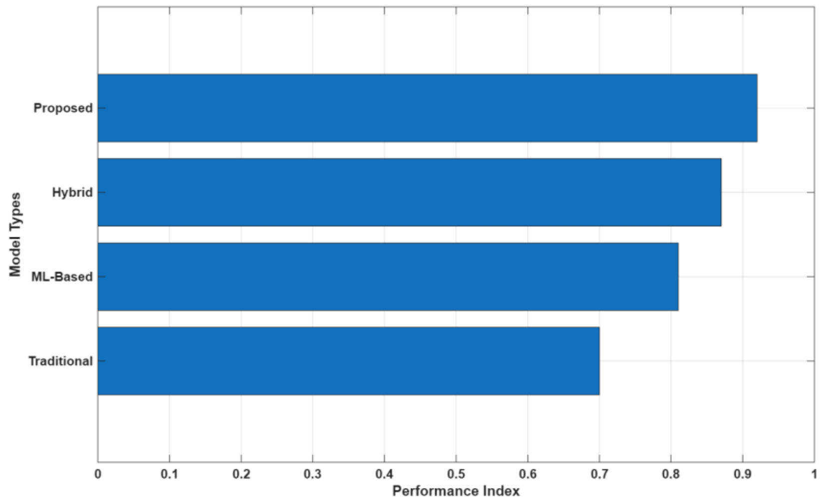

Fig. 9. Performance Index Comparison

Fig. 9 shows a relative analysis of the performance index of the various models. The proposed method scores the highest of 0.92, evidently outperforming all the methods used as baselines. The performance index for hybrid models is 0.87, machine learning-based models are 0.81, and traditional methods lag behind at 0.70. The results indicate that the suggested method is more effective at preserving efficiency and stability under different conditions. This enhancement translates to improved resource control and management, increased response rates, and greater operational consistency, rendering it highly applicable to real-time implementation in distributed and networked systems.

8 Conclusion

The distributed AI framework is suggested to outperform traditional, ML-based, and hybrid approaches in accuracy, efficiency, and response time, with 92.6% accuracy, 89.4% efficiency, and latency down to 182 ms. This is enhanced by the separation of work between edge devices and cloud systems, resulting in faster data processing and reduced communication delays. The system can effectively manage real-time data under higher load and with variable network conditions while still providing stable, consistent performance. It is also leaner in terms of resource usage, eliminating overloading on a single node, and facilitates easy use across distributed components. In total, the framework offers a simple and trusted solution for cyber-physical systems with good scalability, stability, and practical use, with latency and distribution node coordination potentially improved in the future.

References

[1] Rajeh, W., Aborokbah, M. M., S, M., Alashoor, T., & P, K. (2025). TabNet-SFO: An Intrusion Detection Model for Smart Water Management in Smart Cities. *International Journal of Intelligent Systems*, *2025*(1), 6281847. https://doi.org/10.1155/int/6281847

[2] Rao, C. K., Sahoo, S. K., & Yanine, F. F. (2025). A comprehensive review of smart energy management systems for photovoltaic power generation utilizing the internet of things. *Unconventional resources*, *7*, 100197. https://doi.org/10.1016/j.uncres.2025.100197

[3] Dayarathne, M. A. S. P., Jayathilaka, M. S. M., Bandara, R. M. V. A., Logeeshan, V., Kumarawadu, S., & Wanigasekara, C. (2025). Mitigating cyber risks in smart cyber-physical

power systems through deep learning and hybrid security models. *IEEE Access*. doi: 10.1109/ACCESS.2025.3545637.

[4] Kaif, A. D., Alam, K. S., Das, S. K., Chen, G., Islam, S., & Muyeen, S. M. (2025). Blockchain-integrated cyber-physical smart meter design and implementation for secured energy trading in virtual power plants. *IEEE Transactions on Automation Science and Engineering*. doi: 10.1109/TASE.2025.3566938.

[5] Wang, X., Wang, P., Huang, R., Zhu, X., Arroyo, J., & Li, N. (2025). Safe deep reinforcement learning for building energy management. *Applied Energy*, *377*, 124328. https://doi.org/10.1016/j.apenergy.2024.124328

[6] Xie, W., Sun, G., Li, J., Wang, J., Du, H., Niyato, D., & Dobre, O. A. (2025). Generative AI for energy harvesting Internet of Things network: Fundamental, applications and opportunities. *IEEE Internet of Things Magazine*, *8*(3), 72-80. doi: 10.1109/IOTM.001.2400125. doi: 10.1109/IOTM.001.2400125.

[7] Xu, H., Zhang, A., Wang, Q., Hu, Y., Fang, F., & Cheng, L. (2025). Quantum reinforcement learning for real-time optimization in electric vehicle charging systems. *Applied Energy*, *383*, 125279. https://doi.org/10.1016/j.apenergy.2025.125279

[8] Miao, Z., & Liao, Q. (2025). IoT-based traffic prediction for smart cities. *IEEE access*. doi: 10.1109/ACCESS.2025.3552276.

[9] Liu, Y., Han, G., Lin, C., & Zhu, S. (2025). Active Data Routing Based on Reward Backpropagation-enabled Multi-Agent Q-learning towards SDN-enabled Wireless Buoy Networks. *IEEE Transactions on Mobile Computing*. doi: 10.1109/TMC.2025.3611873.

[10] Liu, Y., Meng, Q., Chen, K., & Shen, Z. (2025). Load-aware switch migration for controller load balancing in edge–cloud architectures. *Future Generation Computer Systems*, *162*, 107489. https://doi.org/10.1016/j.future.2024.107489

[11] Nathiya, N., Rajan, C., & Geetha, K. (2025). A hybrid optimization and machine learning based energy-efficient clustering algorithm with self-diagnosis data fault detection and prediction for WSN-IoT application. *Peer-to-Peer Networking and Applications*, *18*(2), 13. https://doi.org/10.1007/s12083-024-01892-8

[12] An, Z., Zhang, L., Fan, Y., Li, Q., & Li, D. (2025). A comprehensive review of TMR current sensors for smart grids: Materials, optimization methods and applications. *Sensors and Actuators A: Physical*, *382*, 116174. https://doi.org/10.1016/j.sna.2024.116174

[13] Wang, Y., Zhang, P., Xie, Y., Chen, L., & Li, Y. (2025). Toward explainable flood risk prediction: Integrating a novel hybrid machine learning model. *Sustainable Cities and Society*, *120*, 106140. https://doi.org/10.1016/j.scs.2025.106140

[14] Umar, H. G. A., Yasmeen, I., Aoun, M., Mazhar, T., Khan, M. A., Jaghdam, I. H., & Hamam, H. (2025). Energy-efficient deep learning-based intrusion detection system for edge computing: a novel DNN-KDQ model. *Journal of Cloud Computing*, *14*(1), 32. https://doi.org/10.1186/s13677-025-00762-9

[15] Sajjadi, P., Dinmohammadi, F., & Shafiee, M. (2025). Machine learning in prognostics and system health management of cyber-physical systems: A review. *IEEE Access*. doi: 10.1109/ACCESS.2025.3605553.

[16] https://www.kaggle.com/datasets/ziya07/iot-smarthome-energy-dataset

[17] https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption