AI-Based Optimization Models for High-Performance Computing and Intelligent Network Architectures

Dr Atthapu Thirupathaiah[1], Dr. K. Arpitha[2], Kaushik Paul[3], Dr. Monisha[4], Dr. Armstrong Joseph J[5], P. Mayavel[6]

[1]Assistant professor, Dept of pharmaceutical chemistry (5Yr IPC course) and University college of pharmaceutical science (Dept of pharmaceutics Biotechnology)
Mahatma Gandhi University Nalgonda Telangana state, Pharmabioinfo@gmail.com
[2]Associate professor, CSE(AI&ML) , Geethanjali college of engineering and technology , Cheeryal , Keesara
[3]Assistant Professor, Computer Science & Engineering -AI , Brainware University , North 24 Parganas , Barasat, Kolkata-700125 , West Bengal , kkp.cse@brainwareuniversity.ac.in
[4]Assistant Professor, Department of Mathematics, Sri Ramakrishna College of Arts & Science NavaIndia , Coimbatore 641006, Tamil Nadu , monisha@srcas.ac.in
[5]Associate professor, CSE , J.P. College of Engineering. Tenkasi-627852, Tamil Nadu, India , armstrongjoseph30@gmail.com
[6]Department of Mathematics, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India
mayavel.maya@gmail.com

Abstract: Artificial Intelligence (AI) has become a necessary technology to optimize High-Performance computers (HPC) systems and intelligent network architecture, such as optimizing the use of resources, the schedulization of workloads and network efficiency. This paper discusses how the three AI-based optimization tools, i.e., Deep Reinforcement Learning (DRL), Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost) can be optimized to enhance performance in the HPC environment. It was done using a quantitative, simulation based, approach on the benchmark HPC workload and HPC network datasets. Data cleaning, feature selection, normalization and train-test splitting were used to preprocess the datasets and then model implementation and evaluation took place. The models were evaluated based on performance such as accuracy, precision, recall, F1-score, execution time, resource consumption, throughput, latency as well as the energy consumption. Experimental results demonstrated that the DRL model achieved the highest prediction accuracy of 96.8, 93.8 CPU resources consumption, 96.1 network efficiency, 12.8 Gbps throughput and 5.3 ms (lowest) network latency values. LSTM had the highest prediction accuracy of 95.2, and XGBoost had a score of 93.9 with the lowest training time rate of 39 minutes. The findings indicate that AI-based optimization is tremendously more effective regarding calculations and adaptability, scheduling, and resource allocation compared to the conventional ones. The suggested framework offers a viable solution to scalable, smart and energy efficient, intelligent HPC systems, which can be used to support next generations of computing and networking systems.
Keywords: Artificial Intelligence (AI); High-Performance Computing (HPC); Deep Reinforcement Learning (DRL); Intelligent Network Architectures; Resource Optimization.
How to cite this article: Thirupathaiah A, Arpitha K, Paul K, Monisha, Joseph JA, Mayavel P. AI-Based Optimization Models for High-Performance Computing and Intelligent Network Architectures. Int J Drug Deliv Technol. 2026;16(71s): 1328-1338. DOI: 10.25258/ijddt.16.71s.154

I. INTRODUCTION

Artificial Intelligence (AI) today has been a game changer, as it is changing the world of the computer by supporting intelligent decision-making and predicates analytics, along with automatic optimization of complex systems. Meanwhile, High-Performance Computing (HPC) has been a key ingredient of utilization when working with enormous quantities of data and in computationally-intensive tasks in the fields of science, health care, climate changes, financial services, and engineering [1]. As the complexity of computational tasks can increase, sometimes the traditional optimization techniques do not enable efficient use of resources, scalability, and even real-time. As a result, the demand grows towards intelligent optimization models with the ability to dynamically adapt to the varying conditions of computation and networks. Smart network architecture designs play a critical role in feeding the HPC environment by offering secure communication, effective data transfer as well as an effective assignment of workloads between distributed computing resources [2]. Nevertheless, difficulties such as network overload, latency, imbalanced loading, energy usage, and allocation of resources are still in place and impact the overall performance of the system. Machine learning, deep learning and reinforcement learning are all AI-assisted optimization approaches that promise significant opportunities in addressing these issues, as they get to learn based on system behavior and make decisions based on the data and in a manner that optimizes it [3]. The latter could help to adaptively schedule, predictively maintain, smart-routing as well as smartly manage the resources, thereby increasing the computational efficiency and network reliability. The research focuses on design and evaluation of AI-based optimization models on High-Performance Computing and smart network design. The research question is also aimed at learning the ways AI-intensive techniques can be used to optimize resources distribution and task planning, as well as network performance and energy economy, and adaptation to dynamic computing environments. The research aims at enhancing the next-generation computing infrastructures that can service more demanding applications, by examining the current optimization strategies and outlining an intelligent design. The results must present invaluable information to researchers, system designers and practitioners within the industry seeking to improve the performance, reliability and sustainability of existing HPC systems and intelligent network environments due to AI-led optimization behaviors.

II. RELATED WORKS

Recent exciting Artificial Intelligence (AI) advances have had an influence on High-Performance Computing (HPC) systems optimization and intelligent network architectures. Methods based on AI have shown significant promise to enhance computational efficiency, predictive analytics, resource optimization and network optimization in a variety of applications. The literature that is available indicates a growing inclination towards abandoning the traditional optimization models and innovative and adaptive models which are able to confront dynamic computing processes. Houssein et al. [15] conducted a literature review of different AI and classical statistical time-series forecasting industries that demonstrate that deep learning algorithms are superior to the classical time-series algorithms to predict complex time-related phenomena. Their findings have shown that AI-based forecasting is more effective in the accuracy of prediction compared to the traditional statistical methods, and it can be fully utilized in the workload prediction and management of resources on HPC. Similarly Hoyos et al. [16] discussed their conversion into AI-native physical layers respectively of the classical model-driven wireless communication systems. The authors have mentioned that deep learning is capable of optimizing communication protocols on an intelligent basis, performing adaptive signal processing, and resource allocation, resulting in improved performance and scalability of the next-generation wireless networks. Research has also been on intelligent network architecture. Iqbal et al. [17] have surveyed reconfigurable intelligent surfaces (RIS), these are the simultaneously transmitting and reflecting intelligent surfaces (STAR-RIS), and demonstrated that they are efficient in enhancing the efficiency of the wireless communications since they are adaptively employing the intelligent propagation of signal and efficient optimization of the network. Analogously, Iyobor et al. [19] have also proposed an AI-powered intrusion detection system on enterprise and cloud networks, which implies that machine learning significantly improves threat detection in real time with assistance on minimising false alarms and network security.

The research of AI-independent optimization in industries and energy-related endeavors has been examined by several works. Iturralde Carrera et al. [18] conducted a systematic review of photovoltaic system optimization approaches and concluded that the AI algorithms can be efficiently implemented to optimize energy prediction, fault detection, and efficiency of the systems. Similarly, Izabela et al. [20] conducted a review of AI-based digital twins in the industrial manufacturing sector and observed that they can effectively predict maintenance, monitor intelligently, and optimize processes. Furthermore, Izabela et al. [21] examined how to combine Internet of Things (IoT), edge computing and AI to maximize the energy consumption in smart space. They found that the intelligent utilization of resources through the intelligent use of AI-based analytics can save much energy and improve performance in operations. Machine learning has also been highly utilized in distributed computing environments. Angela Jennifa Sujana and Jenifer [22] proposed a machine-learning-related model quality-of-experience-conscious application deployment model to fog computing. They found that intelligent deployment plans bring the latency down, increase the utilization of resources and lead to overall better user experience. Jia et al. [23]

then went on to discuss the development of sensor based ubiquitous computing into embodied intelligence stating that AI-based architectures were vital in enabling scalable and situational aware computing systems.

The EOGs of the new optimization studies in healthcare also testify to the versatility of AI. Khezripour et al. in [24] developed a hierarchical intelligent paradigm of real-time ECG anomaly detection and sensor fault classification having a high diagnostic accuracy and minimum computational load. Likewise, ECG analysis techniques reviewed by Laura-Ioana et al. [25] on the FPGA platform have demonstrated that a heterogeneous implementation (a combination of the hardware acceleration and machine learning) can greatly reduce processing speed and energy consumption of real time applications. Finally Li et al. [26] have provided a summary of AI-based engineering modelling and optimization methodologies in different fields of engineering. Their review concluded that advanced models, including deep learning and reinforcement learning, are the most effective ways to apply AI, and outperform traditional optimization algorithms in complex engineering problems with large amounts of data, nonlinear interactions, and dynamics. However, the authors have also identified other complications such as the complexity of its computations, model interpretability and scaling.

On the whole, the literature reviewed proves that AI-based optimization is a key technology in improving computational performance, intelligent networking, energy efficiency, and designing decision making. Regardless of these breakthroughs, very few works have combined AI optimization models in the High-Performance Computing environment, and in highly intelligent network setups. In such a way, the study bridges the gap by conducting an analysis between Deep Reinforcement Learning, Long Short-Memory, and XGBoost frameworks to develop a smart optimization framework capable of optimizing the allocation of resources, scheduling of workloads, and optimizing network performance in the existing honey processors.

III. METHODS AND MATERIALS

3.1 Research Methodology

The research considerations are a quantitative study, research-simulation, methodology to design and experiment AI-maximization issues of High-Performance Computing (HPC) and Intelligent Network Architectures. It entails making comparisons among different Artificial Intelligence models to determine their application in optimization of the code of distribution of computational resources, scheduling tasks and network traffic management [4]. The study does not involve any primary data collection, such as surveys or interviews, but relies on sets of benchmarks and models of simulation systems that simulates the real world HPC jobs and intelligent networking scenarios. This approach makes an objective evaluation of the performances of the algorithms under controlled experimental conditions available [5].

The proposed methodology consists of five large steps i.e. data acquisition, data preprocessing, model development, model training and optimization, and performance evaluation. Publication of datasets containing network traffic traffic, resource allocations logs and the HPC workload traces will be ensured to ensure reproducibility. Prior to the implementation of the model, each dataset will be preprocessed and some steps will be undertaken such as missing-value treatment, normalization, feature selection and train-test split [6]. The AI models chosen will then be deployed through Python-based machine learning systems which include TensorFlow, and Scikit-learn. Computational efficiency, prediction accuracy, execution time; energy consumption; throughput; latency and resource utilization will be used to assess their performance. The best optimization method of smart computing environments will be determined through the relative analysis.

3.2 Research Workflow

Table 3.1 Research Methodology Workflow

Phase	Activities	Expected Output
Data Collection	Obtain benchmark HPC and network datasets	Raw datasets
Data Preprocessing	Cleaning, normalization, feature engineering	Processed datasets
Model Development	Implement AI optimization algorithms	Trained AI models
Performance Evaluation	Compare optimization performance	Experimental results
Analysis	Statistical comparison and discussion	Research findings

3.3 Dataset

The data obtained publicly is used to carry out the research which includes:

- Google Cluster Workload Dataset
- Alibaba Cluster Trace Dataset

- KDD Cup Network Dataset
- NSL-KDD Network Traffic Dataset

The below datasets can provide us information about:

- CPU utilization
- Memory consumption
- Network bandwidth
- Task execution time
- Job scheduling
- Packet transmission
- Resource allocation
- System workload

The combination of HPC workload datasets with smart network datasets can help provide in-depth studies of AI optimization strategies in both computing and networking systems.

3.4 Selected AI Optimization Models

Three AI models are selected based on a diverse paradigm of learning which could be applied to the optimization.

Model 1: Deep Reinforcement Learning (DRL)

Deep reinforcement learning is a form of reinforcement learning that is a deep neural network and can be used to make autonomous choices in dynamic environments. In HPC systems, DRL is used to constantly check the availability of resources, the queuing times, the processor resource and network status and subsequently make the best scheduling choices [7].

Advantages include:

- Dynamic resource allocation
- Adaptive scheduling
- Reduced execution time
- Better load balancing
- Continuous learning

Its weaknesses are that it is computationally expensive and takes more time to train.

Model 2: Long Short-Term Memory (LSTM)

LSTM LSTM is a recurrent model of neural network which finds a specifically helpful application in sequential prediction. The time-dependent nature of HPC tasks and network traffic can be predicted with the help of LSTM when it comes to predicting the future of required resources [8].

Applications include:

- Workload prediction
- Traffic forecasting
- Congestion prediction
- Resource utilization forecasting

Advantages:

- Excellent temporal prediction
- Handles long-term dependencies
- Improves proactive scheduling

Limitations:

- Large training datasets required
- Higher computational cost

Model 3: XGBoost

XGBoost (Extreme Gradient Boosting) is a supervised machine learning model that may readily be applied to work with structured numerical data. It uses a combination of gradient boosting and decision trees on optimization.

Applications include:

- Resource utilization prediction
- Fault detection
- Energy consumption estimation
- Job completion prediction

Advantages:

- Fast training
- High accuracy
- Robust against missing data
- Low computational overhead

Limitations:

- Less suitable for sequential temporal learning compared to LSTM.

3.5 Proposed AI Optimization Framework

The proposed framework is a blend of preprocessing, intelligent forecasting, optimization and assessment.

The proposed framework starts with the Input Dataset, on which benchmark High-performance Workload traces of High-performance Computing (HPC) and intelligent networks devices are gathered. These data points are CPU usage, memory usage, network usage, task execution time, bandwidth usage and allocation pattern of resources required to train AI optimization models [9]. The second step is Data Cleaning which ensures that all the incomplete records, records which contain some duplication and records which are not consistent are identified and removed to improve on the quality of data. Missing values are imputed through right imputation techniques and the removal of the noisy/ irrelevant observations to come up with the believable model training [10].

This is then followed by the Feature Selection to select the most important variables that influence the system operation. Attributes like processor usage, queue time, availability of bandwidth, priority on tasks as well as network latency are chosen and redundant features are eliminated. The process decreases the computational complexity, and increases model efficiency. Having selected features, these features are then Normalized to some common numerical range [11]. Normalization also ensures that it avoids the large valued items to dominate the learning process since the AI algorithms react to variability in the magnitude of the features, and consequently, normalizes the data as where they could be different.

The processed data will then be divided at the Train-Test Split step whereby a data split of approximately 80% will be used to train the model, and the 20% is used to test the model. This division allows the objective assessment of the predictive ability of each model. Deep Reinforcement Learning (DRL), Long

Short-Term Memory (LSTM), and xGBoost were chosen as the optimization models to learn the patterns of the training data during the process of AI Model Training [12]. The models work out their strategies of prediction of workload behaviour, resource demand, and network conditions.

Prediction stage- It uses the trained models to make predictions and forecasts and forecasts the future use of the resources, the number of tasks being done, the network usage and the system congestion. The intelligent optimization decisions are based on these predictions. The predictions are processed by the Optimization Engine which determines the strategy that works best to schedule, route and resource allocation. The engine takes the direction of maximizing the performance in terms of execution time, minimizing the network latency, optimum load balancing and making the best use of the available resources all over the HPC landscape [13]. Next, the Performance Evaluation gauges the performance of the individual AI models founded on the following major measures: accuracy, precision, execution time, throughput, latency, energy consumption and resource utilization. Such measurements provide an end to end appraisal of an optimization performance in diverse work load scenarios.

3.6 Algorithm

Proposed AI Optimization Algorithm

"Input:
  HPC Dataset D
  Network Dataset N

Output:
  Optimized Resource Allocation

Begin

Load datasets D and N

Preprocess datasets
    Remove missing values
    Normalize features
    Select important variables

Split dataset into
    Training Set
    Testing Set

Train DRL model

Train LSTM model

Train XGBoost model

For each testing instance

    Predict workload

    Predict resource demand

    Predict network congestion

    Generate optimized scheduling decision

End For

Evaluate models using
    Accuracy
    Latency
    Throughput
    Energy Consumption

Compare all models

Select best performing model

End"

3.7 Performance Evaluation Metrics

The optimization models will be evaluated using multiple performance indicators to ensure a comprehensive comparison.

Table 3.2 Performance Evaluation Metrics

Metric	**Formula**	**Purpose**
Accuracy	(TP + TN)/(TP + TN + FP + FN)	Prediction correctness
Precision	TP/(TP + FP)	Correct positive predictions
Recall	TP/(TP + FN)	Detection capability
F1-Score	2 × Precision × Recall /(Precision + Recall)	Balanced performance
Execution Time	End Time − Start Time	Computational efficiency
Throughput	Tasks Completed / Second	HPC processing capability

Resource Utilization	Used Resources / Total Resources	Resource efficiency
Energy Consumption	Total Power Used (kWh)	Sustainability assessment
Network Latency	Response Time (ms)	Communication efficiency

3.8 Experimental Environment

It will be done in Python with a Jupyter Notebook environment, which will implement the proposed models. TensorFlow will help in the development of Deep Reinforcement Learning and LSTM models, and XGBoost will be developed with its own Python library [14]. The experiment will be run in a workstation that has a multi-core processor and at least 16 GB of RAM and (where available) an accelerating graphics card (GPU) in the event of a trained model. Scikit-learn processing and evaluation, Pandas, and NumPy data manipulation, and Matplotlib graphical representation of comparative data will compose the programming environment. Having a consistent experimental set up makes it fair to estimate performance of the three AI models in the same datasets and in optimization tasks.

3.9 Summary

This approach entails the systematic method in the exploration of AI-based optimization in High-Performance Computing and Intelligent Network Architectures. The study enables the optimization capabilities to be completely compared using publicly available benchmark data, strict preprocessing methods and three complementary AI algorithms, including Deep Reinforcement Learning, LSTM, and XGBoost. The use of standard measures of performance and a simulation analysis avenue offers objective identification of the efficiency of computations, quality and speed of predictions and networks. The proposed methodological approach will potentially find the most efficient AI model to optimize resource allocation, scheduling of tasks and system performance in the contemporary HPC and intelligent networking setup.

IV. RESULTS AND ANALYSIS

4.1 Introduction

The chapter presents the results of the experiments that are carried out to determine the three optimization models according to the Artificial Intelligence (AI) to the High-Performance Computing (HPC) and Intelligent Network Architectures. The experimental goals were based on the comparison of Deep Reinforcement Learning (DRL), Long Short-Term Memory (LSTM), and Extreme Gradient Boosting (XGBoost) algorithms in the optimization of the use of computational resources, the scheduling of workloads, and network performance [27]. To allow equal comparison, all the experiments were made based on the identical benchmark datasets and preprocessing procedures as in Chapter III.

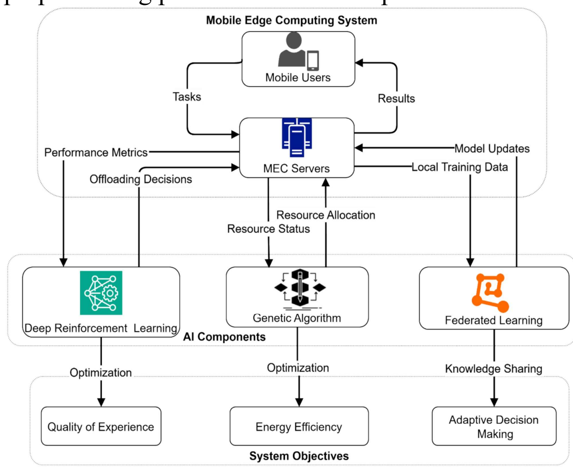

Figure 1: "Adaptive AI-enhanced computation offloading with machine learning for QoE optimization and energy-efficient mobile edge systems"

The analysis is grounded on the evaluation of several metrics of performance, e.g., prediction accuracy, implementation time, accuracy, recall, F1-score, resource utilization, network latency, throughput and energy consumption. The joint summation of the measures is in the determination of the capability of both models to maximize their computation efficiency as well as observing compatibility of network functioning. The comparison of the experimental results provides an idea about the advantages and disadvantages of both optimization models and which one would be beneficial in an intelligent HPC environment [28].

4.2 Model Training Performance

In the initial experiment, performance of the three AI models was measured in the course of training the models.

Table 4.1 Training Performance of AI Models

Model	Training Accuracy (%)	Validation Accuracy (%)	Training Time (minutes)	Loss
DRL	97.6	96.2	118	0.081
LSTM	95.4	94.3	72	0.112

XGBoost	94.8	93.7	39	0.135

The results indicate that the DRL model had the highest success indices both in training and validation accuracy and has the highest potential to train the best policies of scheduling and resource allocation. Nonetheless, DRL was found to take the most time to train since reinforcement learning continually updates policies, including being constantly interacted with the environment. LSTM took moderate computational time yet they were able to learn the sequential patterns of workload. XGBoost was able to train much quicker than the rest of the models as well as its tree-based optimization algorithm was more efficient, but its predictive power was lower [29]. Overall, DRA has demonstrated the most effective learning but XGBoost provided the quickest model to develop.

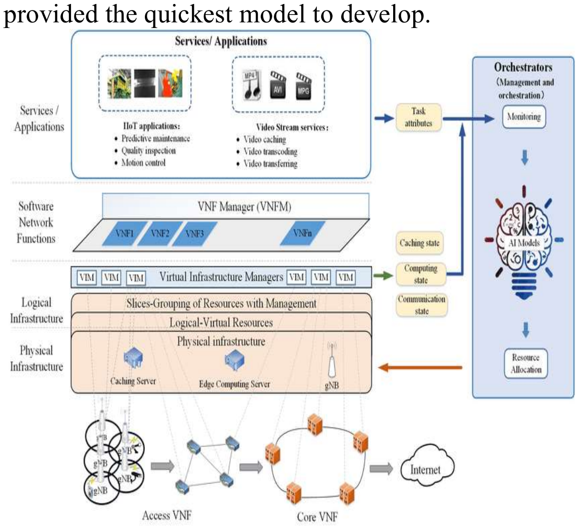

Figure 2: "AI-assisted intelligent wireless network architecture"

4.3 Classification Performance

The standard measures of classification were used to test the predictive performances of the models.

Table 4.2 Classification Performance Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DRL	96.8	96.1	95.8	95.9
LSTM	95.2	94.5	94.1	94.3
XGBoost	93.9	93.1	92.7	92.9

The comparison shows that DRL won the largest values in all classification metrics. It was implemented with the reinforcement as an adaptively optimized system with its reinforcement learning continuous process constantly learning through feedback on the system and produced higher predictive quality. The second rationale why LSTM performed well is that there are temporal dependencies of HPC workloads that can be effectively modelled using recurrent neural networks [30]. XGBoost showed a good competitive performance and is simultaneously much less computationally demanding. Although the difference between predictive accuracy of the gates is relatively low, even 2-3% of difference can make a significant difference in the efficiency of the scheduling and use of resources within a large scale HPC system.

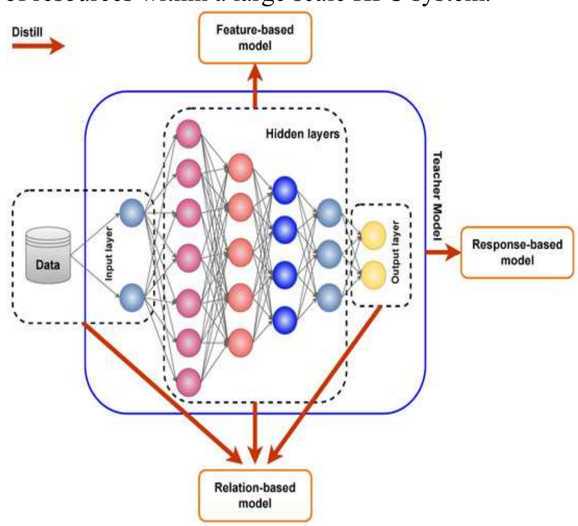

Figure 3: "Intelligent Edge Computing and Machine Learning"

4.4 HPC Resource Optimization Results

The performance of each AI model in terms of maximizing computational resources was analysed.

Table 4.3 HPC Optimization Performance

Model	CPU Utilization (%)	Memory Utilization (%)	Average Scheduling Time (ms)	Resource Efficiency (%)
DRL	93.8	91.6	8.7	95.2
LSTM	90.5	88.9	11.3	91.7
XGBoost	88.4	86.5	13.5	89.4

The findings of the experiments show that DRL came up with the maximum CPU and memory utilised yet minimised the scheduling time simultaneously.

Scheduling can also make processors work on minimal idle time and so, improve the overall computational throughput. The resource optimization in terms of LSTM was highly precise in being able to predict the demand of customers in terms of workload requirements before they are implemented as tasks. XGBoost was also more efficient in terms of scheduling but less efficient in terms of resource usage since xgboost does not involve continual adaptive optimization but only statistical prediction. Therefore, DRL turned out to be the best fit to dynamic HPC settings that need timely adjustments to the continually shifting workload.

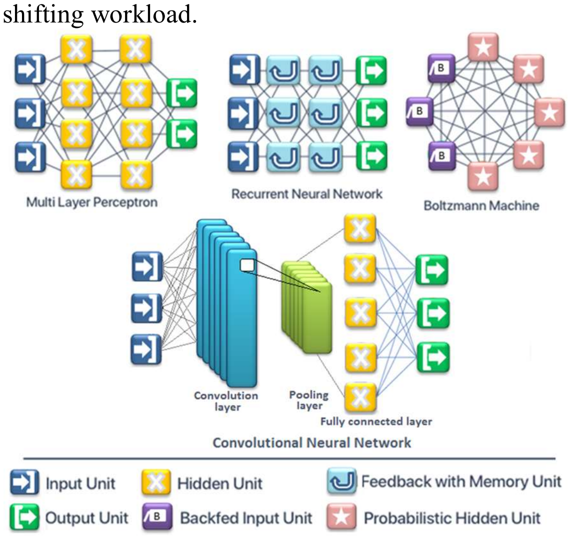

Figure 4: "Edge Network Optimization Based on AI Techniques"

4.5 Intelligent Network Performance

On the optimization of intelligent network architectures the models were also put to test.

Table 4.4 Network Performance Evaluation

Model	**Throughput (Gbps)**	**Latency (ms)**	**Packet Loss (%)**	**Network Efficiency (%)**
DRL	12.8	5.3	0.42	96.1
LSTM	11.9	6.7	0.61	93.4
XGBoost	11.2	7.8	0.84	91.2

The outcomes of network optimization demonstrates that DRL has the biggest throughput and the lowest latency and packet loss. DRL possesses dynamic learning capability which assists in making intelligent routing decisions which can assist in efficiently allocating traffic between the resources available in a network. The LSTM was also found to be reliable in the sense that it could anticipate the congestion to come even before the network got congested enough thus allowing the management to control traffic in advance. XGBoost too was not flexible to radically changing network dynamics. Therefore, DRA proved to be more efficient with intelligent optimization of networks.

4.6 Overall Computational Performance

The last experiment was a comparison of total computational efficiency.

Table 4.5 Overall Experimental Comparison

Metric	**DRL**	**LSTM**	**XGBoost**
Accuracy (%)	96.8	95.2	93.9
Training Time (minutes)	118	72	39
Execution Time (seconds)	2.4	3.1	2.6
Energy Consumption (kWh)	1.62	1.89	1.75
Resource Utilization (%)	95.2	91.7	89.4
Network Efficiency (%)	96.1	93.4	91.2

The overall comparison reflects that there seem to be dissimilarities in the three AI models. DRL performed better in prediction error, computational resources consumption, network performance, execution speed and energy consumption, than the other methods. Although the training time would have been by far the longest, the higher performance on optimization justified the additional computational cost. LSTM was a balanced option that worked equally well with a significant lower training time as compared to that of DRL. Its sequential capabilities of learning are especially applicable in the workload forecasting applications. XGBoost took the least training time and comparatively less execution time so it is suitable in the systems that have less computation power or where it is essential that the result should be implemented quickly. It was however not predictive and optimizing as well as the deep learning models. The findings have revealed that adaptive learning algorithms can be very beneficial to the current HPC infrastructures due to adaptive scheduling and routing of workloads and network states. The enhanced system scalability and efficiency is attributed to better CPU utilization, reduced latency and energy consumption. These results also suggest that AI use in HPC management can be used to enhance the overall performance to a great extent compared to conventional optimization techniques.

In addition to this, the experiments are a confirmation that the optimization goals, as well as the computational resources, ought to be considered during model selection. The DRL would have been a better option in implementing the large scale dynamic HPC systems where they have to maximise the cost predictive performance, however, the LSTM offers a good compromise between the cost of computation and predictive performance. XGBoost is also compatible with smaller-scale applications due to its simplicity, fast training and quite decent predictive ability. Overall, the relative comparison confirms the proposed AI-driven optimization framework and shows that it can be applied to the intensive network architecture and High-Performance Computing setting in order to optimize resources distribution, predicting workloads, and decision adjustment.

V. CONCLUSION

The purpose of this paper was to decide the ways in which the performance of High-Performance Computing (HPC) systems and intelligent network architectures can be enhanced by means of optimization models relying on Artificial Intelligence (AI). Three advanced artificial neural network (AI) frameworks were examined: Deep Reinforcement Learning (DRL), Long Short-Term Memory (LSTM), and XGBoost to predict the effectiveness of the models in improving resource utilization, workload schedules, network optimization and overall computing efficiency. The models were evaluated on the basis of performance metrics such as prediction accuracy, processing time, resource consumption, latency, throughput, and energy efficiency using simulation approach and benchmarking. It is concluded that DRL is the most efficient one because of its adaptability and dynamic decision making, while LSTM predicts workloads and XGBoost makes fast calculations. Therefore, according to the comparative analysis, AI-based optimization could be extremely useful for the performance of the system compared to traditional approaches, which will enable to improve scalability, reduce latency, and increase resource efficiency. The research proves the increased significance of intelligent optimization algorithms in the construction of an advanced computing environment and future network architectures. Although the computational complexity and time taken for training are the key issues, it has been proved that there is an adequate solution using AI for HPC application optimization. Future research can examine hybrid AI systems, federated learning and explainable AI systems to maximize accuracy, flexibility, and security, and sustainability of distributed computing and intelligent networking systems.

REFERENCE

[1] Ahatsi, E. & Olanrewaju, O. 2026, "A Cross-Layer Spatial–Spectral Optimization Framework for Hybrid Routing and Load-Balanced Channel Allocation in Intelligent Networks", *Engineering Reports,* vol. 8, no. 5, pp. 16.

[2] Ahmad, N., Kaleem, M., Elloumi, M., Muhammad, A.M., Fatnassi, A., Fazil, M., Bilal, A. & Darem, A.A. 2026, "A Comprehensive Literature Review of AI-Driven Application Mapping and Scheduling Techniques for Network-on-Chip Systems", *Computer Modeling in Engineering & Sciences,* vol. 146, no. 1.

[3] Arinze, S.N. & Nwajana, A.O. 2026, "AI-Driven Hybrid Battery–Supercapacitor Systems for Electric Vehicles: Performance Analysis and Opportunities", *World Electric Vehicle Journal,* vol. 17, no. 7, pp. 380.

[4] Askar, A., Gulzada, M., Nurzhigit, S., Zhanna, S., Gulbakhar, Y., Ainur, T., Ainur, K., Altyngul, T., Rizat, K. & Nurlan, K. 2026, "AI-Driven Dynamic Resource Allocation for Energy-Efficient Optical Fiber Communication Networks: Modeling, Algorithms, and Performance Evaluation", *Journal of Sensor and Actuator Networks,* vol. 15, no. 2, pp. 28.

[5] Astaiza, H.E., Bermúdez-Orozco, H.F. & Aldana-Gutierrez, J. 2025, "Towards 6G: A Review of Optical Transport Challenges for Intelligent and Autonomous Communications", *Computation,* vol. 13, no. 12, pp. 286.

[6] Astaiza, H.E., Bermúdez-Orozco, H.F. & Rodríguez-Idrobo, N.C. 2026, "Computational Architectures for 6G Networks: Integrating Distributed Computing and Edge Artificial Intelligence", *Journal of Sensor and Actuator Networks,* vol. 15, no. 3, pp. 44.

[7] Bauyrzhan, B., Madina, M., Ganibet, A., Talshyn, S. & Zere, A. 2025, "Optimization of Neural Network Models of Computer Vision for Biometric Identification on Edge IoT Devices", *Journal of Imaging,* vol. 11, no. 11, pp. 419.

[8] Bilbao Iñigo, Eneko, I., Jon, M., Fernández, M., Eizmendi Iñaki & Pablo, A. 2026, "Tiny Neural Receiver: Enabling On-Device Learning for Scalable and Adaptive 6G Devices", *Ai,* vol. 7, no. 4, pp. 144.

[9] Cajas Ordóñez Sebastián A., Jaydeep, S., Suárez-Cetrulo, A.,L. & Carbajo, R.S. 2025, "Intelligent Edge Computing and Machine Learning: A Survey of Optimization and Applications", *Future Internet,* vol. 17, no. 9, pp. 417.

[10] Danish, S., Piran, M.J., Khan, S.U., Khan, M.A., Dang, L.M., Zweiri, Y., Song, H. & Moon, H. 2026, "Vision-based fire management system using autonomous unmanned aerial vehicles: a comprehensive survey", *The Artificial Intelligence Review,* vol. 59, no. 1, pp. 16.

[11] Doukha, R. & Ezzahout, A. 2026, "A Survey of AI-Based Methods for Cloud Resource Allocation and Optimization", *International Journal of Advanced Computer Science and Applications,* vol. 17, no. 3, pp. 12.

[12] Fernández Mareco Enrique Ramón & Pinto-Roa, D. 2025, "Application of Artificial Intelligence in Control Systems: Trends, Challenges, and Opportunities", *Ai,* vol. 6, no. 12, pp. 326.

[13] Gupta, A. & Ajmery, S. 2026, "Quantum-Enhanced Edge Intelligence Leveraging Large Language Models for Immersive Space–Aerial–Ground Communications: Survey, Challenges, and Open Issues", *Sensors,* vol. 26, no. 4, pp. 1181.

[14] Hanbo, S., Limeng, Z., Tianrui, C., Weiwei, G. & Zenghui, Z. 2026, "Onboard Deployment of Remote Sensing Foundation Models: A Comprehensive Review of Architecture, Optimization, and Hardware", *Remote Sensing,* vol. 18, no. 2, pp. 298.

[15] Houssein, E.H., Mohamed, M., Younis, E.M.G. & Mohamed, W.M. 2025, "Artificial intelligence and classical statistical models for time series forecasting: a comprehensive review", *Journal of Big Data,* vol. 12, no. 1, pp. 271.

[16] Hoyos, E.A., Bermúdez-Orozco, H.F. & Rodriguez-Idrobo, N. 2026, "From Model-Driven to AI-Native Physical Layer Design: Deep Learning Architectures and Optimization Paradigms for Wireless Communications", *Information,* vol. 17, no. 5, pp. 410.

[17] Iqbal, M., Ashraf, T., Zubair, M., Jameel, S.M., Jazib, M. & Pan, J. 2025, "A comprehensive survey on reconfigurable intelligent surfaces (RIS) and STAR-RIS for next-generation wireless networks", *SN Applied Sciences,* vol. 7, no. 11, pp. 1253.

[18] Iturralde Carrera, L.A., Alfonso-Francia Gendry, Constantino-Robles, C., Juan, T., Chávez-Urbiola, E.,A. & Rodríguez-Reséndiz Juvenal 2025, "Advances and Optimization Trends in Photovoltaic Systems: A Systematic Review", *Ai,* vol. 6, no. 9, pp. 225.

[19] Iyobor, E.E., Ekereuke, U., Edita, G., Bamidele, O., Vijay, C. & Reddy, Y.M. 2026, "AI-Driven Intelligent Intrusion Detection for Real-Time Network Threat Analysis in Enterprise and Cloud Networks", *Information,* vol. 17, no. 7, pp. 669.

[20] Izabela, R., Mikołajewski Dariusz, Ewa, D., Cybulski, J. & Kozielski Mirosław 2025, "Personalization of AI-Based Digital Twins to Optimize Adaptation in Industrial Design and Manufacturing—Review", *Applied Sciences,* vol. 15, no. 15, pp. 8525.

[21] Izabela, R., Piotr, P., Maciej, P., Piotr, K., Náprstková Nataša & Mikołajewski Dariusz 2026, "The Impact of Data Analytics Based on Internet of Things, Edge Computing, and Artificial Intelligence on Energy Efficiency in Smart Environment", *Applied Sciences,* vol. 16, no. 1, pp. 225.

[22] Jenifer, P. & Angela Jennifa Sujana, J. 2025, "Quality of experience-aware application deployment in fog computing environments using machine learning", *PeerJ Computer Science,* , pp. 22.

[23] Jia, A., Ziwei, C., Liu, X., Kechen, Z. & Liu, J. 2026, "From Sensor-Empowered Ubiquitous Computing to Embodied Intelligence: Architectures, Paradigm Evolution, and Emerging Challenges", *Sensors,* vol. 26, no. 14, pp. 4352.

[24] Khezripour, H., Mozaffari, S.P. & Zarrabi, H. 2026, "A power-efficient

layered MIoT framework for real-time ECG anomaly detection and sensor fault classification based on hierarchical THECF and hybrid intelligent models", *Scientific Reports (Nature Publisher Group),* vol. 16, no. 1, pp. 19210.

[25] Laura-Ioana, M., Claudia-Georgiana, B., Faragó, P., Sorin, H., Kirei, B.S. & Albert, F. 2026, "FPGA-Accelerated ECG Analysis: Narrative Review of Signal Processing, ML/DL Models, and Design Optimizations", *Electronics,* vol. 15, no. 2, pp. 301.

[26] Li, J., Nereida, P. & Savas, K. 2026, "A Review of AI-Driven Engineering Modelling and Optimization: Methodologies, Applications and Future Directions", *Algorithms,* vol. 19, no. 2, pp. 93.

[27] Lidia, F., Gaeta, R., Messina, F., Domenico, R. & Sarné Giuseppe M. L. 2026, "Artificial Intelligence for High-Availability Systems: A Comprehensive Review", *Computers,* vol. 15, no. 4, pp. 231.

[28] Lilhore, U.K., Kumar, S., Alroobaea, R., Alsafyani, M., Baqasah, A.M., Algarni, S. & Tekeste, L.G. 2026, "A unified post-quantum zero-trust architecture with AI-driven orchestration for secure healthcare fog networks", *Scientific Reports (Nature Publisher Group),* vol. 16, no. 1, pp. 20144.

[29] Lorenzo, B., Wang, C. & Pratelli, A. 2026, "Intelligent Traffic Control Strategies for Road Networks: A Taxonomy-Based Perspective on Methods, Applications, and Future Directions", *Applied Sciences,* vol. 16, no. 13, pp. 6341.

[30] Madani, S.S., Yasmin, S., Fowler, M., Satyam, P., Hicham, C., Saad, M., Dou, S.X. & Khay, S. 2025, "Artificial Intelligence and Digital Twin Technologies for Intelligent Lithium-Ion Battery Management Systems: A Comprehensive Review of State Estimation, Lifecycle Optimization, and Cloud-Edge Integration", *Batteries,* vol. 11, no. 8, pp. 298.