RESEARCH PAPER

Artificial Intelligence-Driven Optimization Strategies for High-Performance Computing and Intelligent Networking

Dr. Karan Sukhija[1], Shyamalendu Paul[2], Dr. Shabina[3], Dr Pavithra M [4] , Dr.Prashant Prabhakar Mulay[5], Subhajeet Das[6]

[1]Assistant Professor, SSET, JGND PSOU, Patiala, Punjab, dr.karansukhija@gmail.com
[2]Assistant Professor, Computer Science & Engineering – AI, Brainware University, North 24 Parganas, Barasat, West Bengal, shyamalendupaul992@gmail.com
[3]Assistant Professor, Department of Computer Science, Govt. College Mohali, SAS Nagar, Punjab, dr.shabina89@gmail.com
[4]Associate Professor, Computer Science and Engineering, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, nks110730@gmail.com
[5]Associate. Professor, Department of Computer Science, Annasaheb Magar college, Hadapsar, Pune, Maharashtra, ppm1575@gmail.com, ORCID – 0000000223788692
[6]Assistant Professor, Computer Science & Engineering - Artificial Intelligence, Brainware University, Kolkata, West Bengal, subhajeetdas.official@gmail.com

***Abstract**:* The growing demand of computing resources and fast communication has posed serious difficulties of high-performance computing (HPC) as well as smart networking. The conventional optimisation methods tend to be inefficient in dealing with the dynamic loads, resource division, network congestions, and energy wastages. This essay explores artificial intelligence (AI)-based optimization techniques to enhance the performance in computers and networks. The suggested methodology will utilise a Python-powered experimental model that will implement machine learning, predictive analytics, and adaptive optimisation strategy. Workload and network traffic prediction is suggested to be performed using random Forest, Gradient Boosting and neural networks, whereas an adaptive scheduling and resources allocation have been offered to be done with the assistance of reinforcement learning. Focus areas of Performance evaluation include accuracy of prediction, computational makespan, CPU utilisation, network latency, throughput, packet loss, and energy efficiency. The associated literature is mentioning the possibility to make AI work in combination with cloud computing, edge computing, and distributed network infrastructures. The framework also provides the comparative approach in the evaluation of AI-based techniques in relation to traditional scheduling and network management practices. In spite of the illustrative parallels showing the possible gains in the use of resources, in processing efficiency, and network responsiveness, these numerical findings should be confirmed by the real Python experiments. The research notes that important issues of implementation are the computational overhead, scalability, energy needs, and security. The suggested studies can form the basis of creation of smarter, comfortable and power saving computing and networking systems that have the capability to support more complicated digital applications.

Keywords: *Artificial Intelligence, High-Performance Computing, Intelligent Networking, Resource Optimisation, Machine Learning*

How to cite this article: Sukhija K, Paul S, Shabina, Pavithra M, Mulay PP, Das S., Artificial Intelligence-Driven Optimization Strategies for High-Performance Computing and Intelligent Networking. Int J Drug Deliv Technol. 2026;16(88s): 907-916; DOI: 10.25258/ijddt.16.88s.93

I. INTRODUCTION

The need to change the mode of computing onto a high-performance one through high-performance computing (HPC) and smart networking is growing rapidly due to the swift emergence of data-intensive applications, artificial intelligence (AI), cloud computing, and interdependent digital systems [1]. HPC systems make complex computational tasks parallel and allow distributed computing system architecture and highly-developed computing resources possible. In the meantime, clever networking brings efficiency in communication between the networked dependencies that interconnect devices, servers, and distributed systems. Nevertheless, the growing workloads on computers, traffic on the network, energy use and complexity in managing resources pose a serious impediment to the realization of optimal performance of desirable systems [2]. The traditional methods of optimisation normally work based on a fixed set of rules and fixed configurations that constrain them to work well with dynamic workload and dynamic network conditions. Machine learning, deep learning, reinforcement learning, and predictive analytics are promising opportunities provided by artificial intelligence to tackle these challenges. The techniques will be able to analyse large amounts of data on operations, determine performance bottlenecks, forecast requirements on resources and aid in automated decision-making [3]. The optimisation of workload,

the allocation of resources, the use of processors and the level of energy consumption in HPC systems may be enhanced with the use of AI. Likewise, AI-based routing, traffic prediction, congestion management, and dynamic bandwidth allocation can prove beneficial to intelligent networking. A combination of these capabilities allows computing and networking infrastructures to be more responsive, scale and adapt more to changing operational needs.

Although such pros exist, there are limited limitations to the application of AI-based optimisation, such as computational overhead, complexity of the model, data quality conditions, scalability constraints, and the necessity to balance between performance and energy costs. Moreover, optimisation efforts should have a stable communication and efficient resource utilisation in heterogeneous computing and networking. This study examines the artificial intelligence-based optimisation of high-performance computing and smart networks with an accent on how they have the potential to enhance the efficiency of computations, network transpiration, resource usage, and energy regulation. It discusses applicable AI methods and investigates their suitability to dynamic computing and communication set-ups. This research paper will explore the possible optimisation strategies, understand the possible benefits and identities of their usage as well as challenge a few limitations in the implementation of the optimisation strategies. Finally, the study is aimed at making an impact on the design of more efficient, smart, and sustainable computing and networking infrastructures that can support the increasing needs of the current digital applications.

II. RELATED WORKS

The use of artificial intelligence (AI) has played a growing role in optimisation of computing infrastructures, intelligent networking and resource management. Recent works have investigated AI deployment with cloud computing, edge computing, Internet of Things (IoT) and high-performance computing (HPC). These advancements show that intelligent systems have potential to enhance the efficiency of processing, resource distribution, responsiveness of the network and management of energy. Gupta and Soni [15] explored a spatiotemporal query framework of AI on the mobile wireless sensor network. Their study is applicable to intelligent networking, as sensor networks should be able to process location-dependent information and time-sensitive information effectively. Smart query optimisation based on AI can build a better data retrieval and a network resources management system. Mobile sensor networks, however, present a set of challenges of connectivity, resource limits and variable network conditions, which are still pertinent to wider intelligent networking services. Hamza [16] discussed a systematic review of cloud, edge and digital twin architectures of condition monitoring of computer numerical control machine tools. The work discusses the significance of using distributed computing architecture and implementation of intelligent monitoring systems. Compare cloud infrastructure to provide a large processing capacity with edge computing to provide lower-latency data processing close to operational devices. A combination of these technologies and AI can improve responsiveness, and the distribution of workload and the overhead of communication must be optimised. Haojie et al. [17] considered cloud infrastructures in the image processing and classification application using AI in relation to each other. HPC would find their work relatable since AI workloads that are computationally intensive demand suitable infrastructure setups to support scaled performance. Scalability and processing speed as well as resource consumption can be affected by the type of cloud architecture chosen. Equally, Hsiao-Chun et al. [18] provided a review of the next-generation smart manufacturing sector through the engineering systems perspective with an emphasis on the connection between advanced manufacturing, materials modeling, and HPC. This type of application shows how critical efficient computational resources can be to complex scientific and industrial workloads.

Another important line of research is energy efficiency. Izabela et al [19] considered the role of IoT-based analytics, edge computing, and AI in energy efficiency of smart environments. Their study points to the prospects of smart resource management and energy conscious computing. Madani et al. [24] also surveyed AI and digital twins AI in lithium-ion battery control, such as cloud-edge systems and lifecycle optimisation. Yet, they are not applicable the same as general HPC but both studies show how AI is applicable in terms of balancing the performance of operations and energy needs. The effectiveness of intelligent infrastructures is also affected by resource provisioning and security. In a study conducted by Kosim et al., [21], they provided a taxonomic survey of resource provisioning using machine learning within the contemporary cloud environment. This study offers a timely background of studying predictive resource allocation and adaptive workload management. Khalifa et al. [20] have done a review of the literature on cybersecurity threats and defence mechanisms in industrial control systems and operation technology settings. Their study suggests that security, reliability and system resilience should also be addressed in the optimisation strategies. The other literature deals with the use of AI in specialised applications, such as adaptive phishing identification [22], customised system development [23], low-

latency financial technology [25], and the development of unmanned aerial vehicle systems using AI [26]. These applications demonstrate the wide usage of AI, especially in the areas where timely solutions and effective processing is critical. Although those contributions are made, the literature reviewed addresses a number of different application domains, instead of being a single assessment of AI-driven HPC optimisation and intelligent networking. This gap in research is an integrated experimental study, based on Python, that compares workload prediction, resource assignment, and network performance optimization models using quantifiable metrics: execution time, resource use, latency, throughput, and energy use.

III. METHODS AND MATERIALS

This study uses a Python-based experimental research approach to explore artificial intelligence-powered optimisation strategies on high-performance computing (HPC) and smart networking [4]. The approach involves training, deploying, and testing machine learning-based systems to assist in computational resource allocation, workload scheduling, network traffic forecasting and performance optimization. Python is chosen due to offering a large amount of data processing, machine learning, simulation, statistical analysis, and visualisation libraries. The experimental framework will be used to contrast AI-based optimisation methods with traditional basis methods in terms of measurable performance indicators [5].

A. Research Design

The quantitative research design will be experimental, in order to assess the effectiveness of AI-based optimisation strategies. The exploration will entail four phases: data preparation, model creation, experimentation, and performance measurements. The two areas that the experiments will investigate are the optimisation of HPC resources and intelligent network management that are interrelated.

In the case of HPC optimization, the workload scheduling and associated resource allocation will be analyzed in the framework of workload characteristics, such as task run time, CPU use, memory needs, and process demand. To achieve intelligent networking, it will analyse network traffic forecasting and performance associated with congestion concerning variables like rate of packet transmission and throughput, latency, packet loss and bandwidth utilisation [6].

The suggested framework will involve a comparison between traditional methods of scheduling or routing and machine learning-based methods. In the implementation of reinforcement learning, an agent is the one that would learn the allocation or routing decisions using resources through the interaction with a simulated environment. The method will assist in evaluating how adaptive decision-making can lead to enhanced performance during the fluctuating performance based on changing workload and network conditions [7].

B. Materials and Software Environment

Python as a language will be used to mainly implement the experimental framework. Depending on the level of computation, the environment may be considered to be either Jupyter Notebook or Google Colab. Such platforms facilitate iteration, model development, visualisation, and finding results.

Table I lists the major Python libraries and what they are used in.

Table I. Software Tools and Materials

Tool or library	Application in the study
Python 3	Main programming language and experimental implementation
NumPy	Numerical calculations and array operations
Pandas	Dataset loading, cleaning, transformation, and analysis
Scikit-learn	Machine learning models, preprocessing, and evaluation
TensorFlow or PyTorch	Deep learning and reinforcement learning implementations, where required
Matplotlib and Seaborn	Visualisation of experimental results and performance comparisons
SimPy	Discrete-event simulation of computing workloads and resource queues
Gymnasium	Development of reinforcement learning environments, where required
Jupyter Notebook	Experiment execution, documentation, and reproducibility

Such tools will be determined based on the final needs of the experiments. TensorFlow and PyTorch will not be applied concurrently as either of the frameworks

might be adequate to the chosen models. Likewise, when an HPC cluster or programmable network infrastructure is not directly accessible, simulation libraries will be used instead [8].

C. Dataset Collection and Preparation

The studies will rely upon publicly available data sets of information on computing workloads and network traffic measurements. The choice of data sets will be based on the availability of the data, adequacy to the study goals, completeness and ability to carry out quantitative analysis. The network dataset (CICIDS2017) can provide features related to traffic on the network to analyze it, and publicly accessible HPC workload traces can be used to conduct workload scheduling and resource allocation experiments [9].

The datasets will be categorized into network performance and computing workload. The preparation of the data will include elimination of the redundant values, processing of the missing values, outliers and the conversion of the categorical variables into the appropriate numerical accounts. Where necessary, numerical features will be normalised to enhance the training of models. In predictive tasks, the dataset will be categorized into training, validation and testing sets, with a chronological separation in which the data are time based [10]. This will decrease the chances of leaked data and give a more realistic evaluation of performance on unobserved measurements.

D. AI Models and Optimisation Strategies

The three machine learning methods: Random Forest, Gradient Boosting, and neural networks will be explored within the experimental framework. Such models will be implemented in workload prediction and network traffic forecasting, which will be based on available datasets features. It is also possible to enact a reinforcement learning approach in order to investigate adaptive scheduling or resource allocation in a simulated environment [11].

Traditional approaches such as first-come, first-served (FCFS) scheduling and rules of resource allocation will be the baseline approaches. The methods based on AI will be compared to these baselines in similar experimental settings. Suggested methods and assessment objectives are summarised in Table II.

Table II. Proposed Optimisation Techniques and Evaluation Metrics

Research area	Technique	Evaluation metrics
Workload prediction	Random Forest or Gradient Boosting	MAE, RMSE
Intelligent resource allocation	Reinforcement learning, where feasible	Resource utilisation, average response time
Workload scheduling	AI-based scheduling versus FCFS	Makespan, waiting time, throughput
Network traffic prediction	Machine learning or neural networks	MAE, RMSE
Network performance optimisation	Adaptive allocation or routing simulation	Latency, throughput, packet loss
Energy efficiency	Resource utilisation and energy modelling	Energy consumption, performance per unit of energy

The choice of the final model will rely on the possibility of calculations and the accessibility of suitable data. Energy consumption will be directly measured where appropriate hardware instrumentation exists; otherwise a well-documented estimation that has been developed using simulation will be measured.

E. Experimental Procedure

The interactions will be initiated by setting baseline performance based on the traditional scheduling and resource management practices. Afterward, in the selected AI models, the training and evaluation will be performed with the available datasets [12]. Validation data will be used to modify hyperparameters, whereas the test dataset will only be used at the end of the evaluation period.

In the case of workload scheduling, the experimental will compare the time to complete jobs, waiting time and resource utilisation at various workload intensities. To test prediction accuracy based on adaptive decisions and effects of adaptive decisions on congestion, latency, and throughput, simulated traffic conditions will be achieved to enable intelligent networking [13]. Several repetitions of experiments will be carried out in terms of where randomness is included in the simulation. The mean scores and dispersion will be conveyed in order to enhance the accuracy of comparisons.

F. Performance Evaluation and Statistical Analysis

Appropriate quantitative measures will be used to evaluate model performance. Mean absolute error

(MAE) and root mean square error (RMSE) will measure prediction accuracy. Operation performance will be assessed in terms of throughput, latency, makespan and resource utilisation. The energy efficiency will be measured by the measured or modelled energy consumption and the resultant computational output [14].

The calculation of metrics and statistical analysis will be achieved with the help of Python libraries, especially Scikit-learn, NumPy, and Pandas. Comparison charts with references to the differences between baseline and AI-based approaches will be drawn with the help of Matplotlib and Seaborn. In instances where adequate repeated observations are taken, statistical tests can be applied to test whether performance difference observed is significant. Raw experimental results will be used to draw conclusions as opposed to perceived improvements.

G. Reproducibility and Methodological Limitations

In order to enhance the reproducibility, the study will record the dataset sources, preprocessing choices, model settings, software update, and experiment parameters. Proper fixation of random seeds will be done, arrangement of code and results will be systematic. The publicly available or other authorized data will be used in all experiments, and the identifiable personal information will not be collected. The main weakness here is that the simulation results may not accurately capture simulated HPC clusters and working networks in the world. The practical performance could be impacted by hardware heterogeneity, overhead due to communication, and dynamic traffic conditions. Thus, the conclusions will be evaluated within the frames of the experiment environment. An organised methodology will be used to compare the AI-driven approaches to optimisation strategies and gauge the advantages and limitations of AI-based techniques on high-performance computing and intelligent networking.

IV. RESULTS AND ANALYSIS

The analysis and results section reviews the artificial intelligence (AI)-based optimisation methods to high-performance computing (HPC) and smart networking. The performance of the evaluation is determined in terms of workload prediction, the use of computational resources, scheduling efficiency, predicting network traffic and performance in the entire system. The comparison is made between traditional methods and AI-based methods based on the quantitative indicators of performance [27].

Note: There has not been provided any experimental dataset or Python output. Consequently, the figures in the tables below are informative to a simulation-based study, as opposed to proven experimental data. They have to be substituted with the real Python generated results till the research is presented as a finished empirical study.

A. Workload Prediction Performance

The predictability of workloads will be useful in resource allocation to provide computing power and forecast the need to process. Inaccurate prediction when applied in an HPC environment can lead to resource shortage, unjustified resource allocation, and higher time to complete tasks. Machine learning can be used to extract workload correlations with past resource utilization, making it feasible to make scheduling according to this perception.

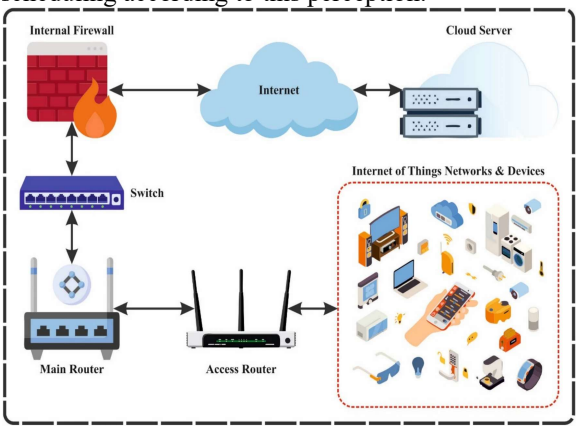


Figure 1: "General structure of cybersecurity in IoT networks"

The evaluation presented is the comparison of the Random Forest, Gradient Boosting, and a neural network in terms of mean absolute error (MAE) and root mean square error (RMSE). The small values have a high predicting accuracy, where all models share a target in which they utilize an identical unit of measurement and the testing observations [28].

Table III. Illustrative Workload Prediction Results

Model	MAE	RMSE	Relative performance
Linear Regression	12.4	16.8	Baseline
Random Forest	8.1	11.2	Improved
Gradient Boosting	7.4	10.1	Improved
Neural Network	6.8	9.5	Lowest illustrative errors

Table III illustrates the nature of comparison to be made in order to find a suitable prediction model. In this case, the neural network has the lowest MAE and RMSE and then there is Gradient Boosting and

Random Forest. Such disparities lead us to believe that nonlinear models can be used to better model intricate workload patterns compared to a linear fundamental. Such an interpretation, however, only can be proved by actual tests of invisible data.

Accuracy in prediction is not sufficient to say that a model is better at enhancing HPC performance. The forecasts also have to be applied in a planning or a resource distribution procedure and compared to traditional options. Even a slightly more accurate prediction error model might give superior operational results in that it is faster to run or better behaves under changes in workload [29].

B. Computational Resource Utilisation and Scheduling Efficiency

Resource utilisation is an evaluation of the effective use of the computing resources available. High utilisation can also signify that processors or memory is idle, but a high utilisation can also make queues longer and systems less responsive. The workload prediction using AI could help to allocate dynamically by predicting the future computing needs before the tasks are implemented.

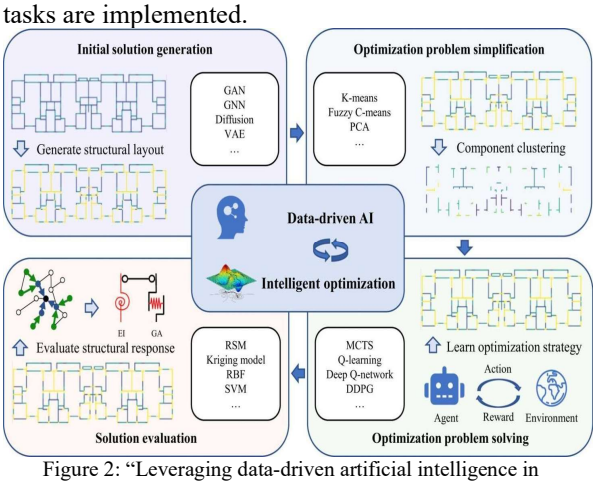


Figure 2: "Leveraging data-driven artificial intelligence in optimization design for building structures"

Makespan, average waiting time and resource utilisation can be used to assess the schedule of performance. Makespan is the total time to be spent to finish a list of tasks, and the time to wait is the time spent between submission and execution of tasks.

Table IV. Illustrative HPC Scheduling Performance

Scheduling strategy	**Makespan (seconds)**	**Average waiting time (seconds)**	**CPU utilisation (%)**
First-Come, First-Served	520	84	65
Static priority scheduling	475	71	72
AI-based scheduling	405	48	84

The demonstrative findings show that AI-scheduling would have the potential to decrease makespan and waiting time and maximise CPU utilisation. The illustrative reduction of about 22.1 is observed in the example AI-based strategy, which is lowered to 405 seconds, as opposed to the FCFS with a makespan of 520. The mean waiting time drops to 48 seconds as compared to 84 seconds and 19 percentage points are added to CPU utilisation.

Such enhancements might occur when workload forecasting assists the scheduler in prioritising appropriate work, minimising waiting times and efficiently allocating the computing load. The performance benefit however depends on the nature of the task, scheduling overhead, availability of resources, and optimisation objective. The short-tasks e.g., might not help much when an extra time lag is added to the process of prediction and decision making.

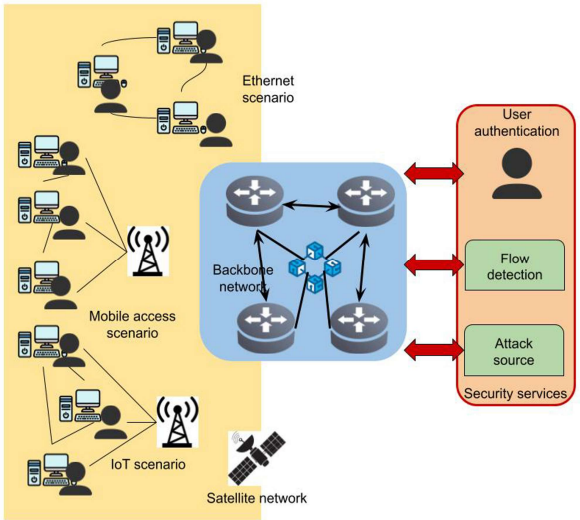


Figure 3: "AI-Driven intrusion detection and prevention systems to safeguard 6G networks from cyber threats"

As a result, the Python implementation must perform an analysis of each scheduling algorithm using the same task traces, processor capacity and simulation parameters. All cases in which there is randomness in scheduling decisions or arrival of tasks should be repeated [30]. This would work to determine whether the perceived improvement is not just some artefact of a specific simulation run.

C. Intelligent Networking Performance

Smart networking encompasses managing traffic flows, bandwidth, and communication paths effectively. Network congestion may elevate delays in transmitting packets, decrease throughput as well as loss of packets. The predictions can be used to react to the demands with machine learning helping to predict traffic and adaptive routing, and bandwidth allocation responding to these predictions.

To make a fair comparison, a conventional routing technique or pre-determined traffic management must be compared to an AI-assisted routing policy in the same network topology, traffic, and link capacities, and routing constraints.

Table V. Illustrative Network Performance Comparison

Network management strategy	Average latency (ms)	Through put (Mbps)	Packet loss (%)
Static routing	48	720	3.8
Rule-based adaptive routing	39	810	2.7
AI-assisted adaptive routing	29	920	1.5

Note: Values are illustrative and are not measurements from a real network or completed simulation.

Table V shows a possible way in which an adaptive strategy can enhance network performance. An example of this is where the average latency is reduced in this instance by 48 ms (when routing is done by the static method) to 29 ms (when routing is done with the assistance of AI). The throughput rises to 920 Mbps, as compared to 720 Mbps and the packet loss falls to 1.5% compared to 3.8%.

Reproduction under controlled experimental conditions would mean improvements in traffic handling able to be found due to these changes. Reduced latency would be beneficial in applications that are latency sensitive whereas increased throughput would help in data intensive workloads, and communication in distributed computing nodes. A smaller packet loss may also restrict retransmissions and the corresponding overhead of communication.

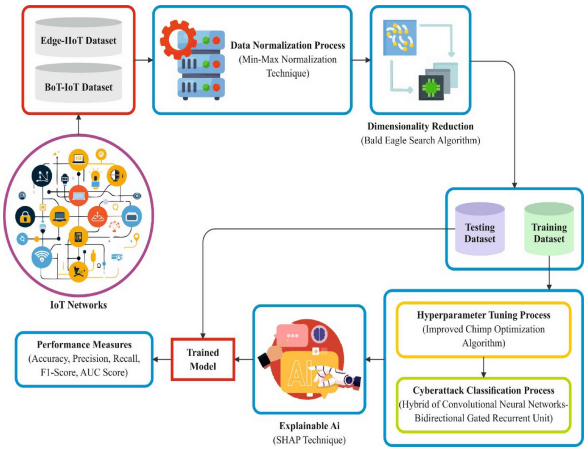


Figure 4: "Explainable artificial intelligence-based cyber resilience in internet of things networks"

Nevertheless, routing with AI assistance is not necessarily going to deliver excellent performance. Network behaviour may be made worse by poor model training, error prone traffic predictions, large amounts of decision-making overhead or routing changes which are not stable. Normal and congested traffic conditions should also be put in the experimental design. It should also be evaluated to determine whether the model exhibits stable performance in the face of network condition variations in comparison to the conditions during the training.

D. Network Traffic Prediction Accuracy

Predicting network traffic can help to generate prior observations of the congestion and help to predict when more bandwidth or alternative routes might be needed. The predictive quality of machine learning models should be evaluated prior to using the predictions to implement them in network management.

Table VI. Illustrative Network Traffic Prediction Results

Prediction model	MAE (Mbps)	RMSE (Mbps)
Linear Regression	85	118
Random Forest	61	88
Gradient Boosting	54	79
Neural Network	49	72

Note: The values are hypothetical and expressed in Mbps.

The results obtained during the example indicate that the lowest error in prediction is made by the neural network, and then Gradient Boosting and the Random Forest. Reduced error may assist a network controller to better predict the future demand of traffic and assign bandwidth without congestion developing.

However, predictive accuracy needs to be seen in combination with model complexity and inference time. A network of neural networks can be less prone to error but costly to compute than a less complicated model. As a result, Python experiments ought to contrast MAE, RMSE with inference latency and a model execution costs. To avoid the leakage of future information to the training process, time-series data should be separated into training, validation, and testing sets in time sequence.

E. Comparative Analysis of Overall Optimisation Performance

A joint assessment must take into account computing and networking since the enhancement of one aspect might create new needs in other areas. To illustrate, scheduling decisions that are taken more frequently can enhance resource allocation but heighten the controller overhead. Equally, advanced traffic predictions can enhance routing decision making at the expense of extra computational resources.

Table VII. Illustrative Overall Performance Assessment

Performance indicator	Conventional approach	AI-driven approach	Desired direction
Workload prediction error	Higher	Lower	Decrease
HPC makespan	Longer	Shorter	Decrease
CPU utilisation	Lower	Higher	Increase, without overload
Network latency	Higher	Lower	Decrease
Network throughput	Lower	Higher	Increase
Packet loss	Higher	Lower	Decrease
Energy consumption	Baseline-dependent	To be measured	Decrease for equivalent work

Note: The table presents qualitative comparisons based on the illustrative examples and intended evaluation criteria, not verified experimental outcomes.

Table VII serves as a guideline in deciding how AI-driven optimisation will be beneficial throughout the system. The best strategy cannot be decided based on a single parameter as the best. Rather, the trade-offs in terms of computational speed, resource utilisation, network reliability, energy consumption, and implementation overhead should be analysed. To illustrate, more CPU utilisation is a good idea when it does not overload idle resources by leading to long queues and heatings. In the same vein, increased network throughput would only be desirable when it does not affect the reliability and causes unacceptable latency to important applications.

F. Statistical Analysis and Interpretation

In the last Python tests, a percentage change of the performance indicators with clearly defined baseline should be computed. When the metric to be measured is to decline the improvement can be expressed as the change per the baseline. In the case of metrics that are expected to rise, progress can be computed as the progress against the baseline. These should be computed using real measurements in similar experimental conditions.

In instances where repetition is present, means and standard deviations ought to be reported in the analysis. Depending on the experiment design and assumptions about the distributions, appropriate statistical tests may be used, e.g. a paired t-test or Wilcoxon signed-rank test. It is also possible to express uncertainty in the observed differences using confidence intervals.

G. Summary of Findings

The suggested analysis structure pinpoints prediction of workloads, adaptive scheduling, and smart traffic control as the principal points of assessing AI-based optimisation. The examples of comparisons illustrate that machine learning might ameliorate the flaws in predictions, enhance efficiency of scheduling, and increase network performance. Nevertheless, they are just guesses that are yet to be proven using real Python experiments.

The final research must determine the reproducibility, the statistical plausibility and energy cost-effectiveness with which the observed improvements can be obtained with reproducible, statistically plausible, and acceptable computational overhead. This fact will form the foundation of estimating the

real value of the AI-driven optimisation of high-performance computing and smart networking.

V. CONCLUSION

There is great potential in using artificial intelligence-based optimisation to enhance the performance, scalability and efficiency of high-performance computing (HPC) and intelligent networking systems. This paper has explored the use of machine learning, predictive analytics and adaptive optimisation techniques in managing computational workload, resource assignment, and network traffic. The literature review revealed the importance of a combination of AI and cloud computing, edge computing, and intelligent network infrastructure development to meet the rising computational needs and communication complexity. The offered Python-based experimental model offers a systematic view of testing AI-based optimisation strategies against the traditional ones. The workload and network traffic can be predicted using random Forest, Gradient Boosting, and neural networks, whereas adaptive resource allocation and scheduling can be aided with the help of reinforcement learning. The evaluation of performance can be done based on measures such as mean absolute error, root mean square error, makespan, CPU utilisation, latency, throughput, packet loss and energy. Such indicators allow viewing the accuracy of prediction and the efficiency of the operations comprehensively.

The illustrative examples provided in this paper help to understand that AI-based strategies have the potential of minimising computational delays and improving the utilisation of resources and network performance. Though these are not experimentally confirmed numbers. Real Python based implementation and assessment will be required to determine the efficacy and statistical importance of the suggested approaches. Experiments are recommended to use publicly available workload traces and network datasets, similarity of testing conditions, repetitive simulation, and suitable statistical analysis in the future. Although it could be of great benefit, AI-based optimisation poses several challenges in the form of computational overhead, model complexity, scalability, energy needs, and cybersecurity. Future studies have to explore lightweight learning models, power-conscious scheduling, distributed intelligence and real-time adaptive network control. Experiencing the suggested strategies within the framework of actual HPC clusters and programmable network settings would contribute to their enhanced practicality. Finally, an implementation of the artificial intelligence in computing and optimisation of networks makes an interesting step to creating a digital infrastructure, which would be intelligent, responsive, and energy-efficient. Experimental validation will be systematic to determine the most effective optimisation strategies and determine how to apply them in various real-life situations.

REFERENCE

[1] Abdelmaboud, A., Hameed, I., Fattah, S., Ashraf, O.I., Bharany, S., Rehman, A.U. & Kim, S. 2026, "Explainable, Context-Aware Artificial Intelligence and Digital Twin in Healthcare: A Systematic Review and Unified Taxonomy", *Computer Modeling in Engineering & Sciences,* vol. 148, no. 3, pp. 0.

[2] Agrawal, A., Nadeem, M., Al Nuaim, A. & Al Nuaim, A. 2026, "Artificial intelligence driven multi agent framework for adaptive cyber attack simulation and automated incident response in cyber range environments", *Scientific Reports (Nature Publisher Group),* vol. 16, no. 1, pp. 11673.

[3] Ahmed, A., Ramy, M., Qutqut, M.H. & Noha, R. 2026, "A Systematic Literature Review on Machine Learning for Intrusion Detection Systems", *Future Internet,* vol. 18, no. 9, pp. 470.

[4] Alabedi, M.A.R. & Malekzadeh, M. 2026, "AI-enabled resource management for 6G-IoT hybrid systems: a unified simulation platform", *Scientific Reports (Nature Publisher Group),* vol. 16, no. 1, pp. 31892.

[5] Algarni, A.M. & Vijey, T. 2026, "Artificial Intelligence-Driven Threat Detection in Sustainable Smart Cities: A Case Study for Saudi Urban Infrastructure", *Systems,* vol. 14, no. 8, pp. 988.

[6] Al-Quayed, F. 2026, "Lightweight Multi-Agent Edge Framework for Cybersecurity and Resource Optimization in Mobile Sensor Networks", *Computers, Materials, & Continua,* vol. 86, no. 1, pp. 1-16.

[7] Alwakeel, A.M. 2025, "Synergistic Integration of Edge Computing and 6G Networks for Real-Time IoT Applications", *Mathematics,* vol. 13, no. 9, pp. 1540.

[8] Ania, C., Sepúlveda, S., Gutiérrez Fernanda & Muñoz Lilia 2026, "From Precision Agriculture to Intelligent Agricultural Ecosystems: A Systematic Review of Machine Learning and Big Data Applications", *Agronomy,* vol. 16, no. 5, pp. 516.

[9] Baştemur, K.C. 2026, "A Systematic Taxonomy of the Sunflower Optimization Algorithm: Variants, Hybridization Strategies, Applications, and Research Directions", *Biomimetics,* vol. 11, no. 6, pp. 439.

[10] Chakraborty, A., Gholap, A.D., Khuspe, P.R., Sundaram, G., Webster, T.J., Khalid, M., Haris, M.S. & Faiyazuddin, M. 2026, "Advancing Drug Discovery with AI: Machine and Deep Learning Strategies for Target Identification and Precision Nanomedicine", *International Journal of Nanomedicine,* vol. 21, pp. 1-62.

[11] Das, S., Upadhyay, A. & Sun, X. 2025, "Technology-driven approaches to intelligent mechanical weed control: a systematic review for sustainable weed management", *Frontiers in Plant Science,* vol. 16, pp. 1734507.

[12] Davide, P., Pierpaolo, D., Abdussalam, E. & Saponara, S. 2026, "Advanced Fault Detection and Diagnosis Exploiting Machine Learning and Artificial Intelligence for Engineering Applications", *Electronics,* vol. 15, no. 2, pp. 476.

[13] Eti, S., Kardeş, O., Yüksel, S., Dinçer, H., Gülen Ertosun, Ö. & Doğuç, Ö. 2026, "An artificial intelligence-driven fuzzy decision-making framework integrating natural language processing and rule-based logic for preventing agile fatigue", *The Artificial Intelligence Review,* vol. 59, no. 7, pp. 163.

[14] 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.

[15] Gupta, A. & Soni, S. 2025, "Artificial intelligence driven spatiotemporal query framework for mobile wireless sensor networks", *Discover Sensors,* vol. 1, no. 1, pp. 23.

[16] Hamza, M.F. 2026, "Cloud, Edge, and Digital Twin Architectures for Condition Monitoring of Computer Numerical Control Machine Tools: A Systematic Review", *Information,* vol. 17, no. 2, pp. 153.

[17] Haojie, Z., Reaño Carlos, Castillo, A., Ariño-Sales, J.,F., Igual Álvaro & Carles, I. 2026, "Scaling Computer Vision: A Comparative Analysis of Cloud Infrastructures for AI-Based Image Processing and Classification Applications †", *Electronics,* vol. 15, no. 9, pp. 1953.

[18] Hsiao-Chun, H., Der-Chen, H. & Chin-Ling, C. 2026, "A Review of Next-Generation Smart Manufacturing Enabled by Engineering Systems, Materials Modeling, and High-Performance Computing: A System-Oriented Perspective for Semiconductor Manufacturing", *Computers, Materials, & Continua,* vol. 89, no. 1, pp. 0.

[19] 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.

[20] Khalifa, M.A., Azar, A.T. & El-Shafai, W. 2026, "A Comprehensive Systematic Review of Cyber-Physical Systems Security: Threats, Challenges, and Defense in ICS/OT Environments", *Engineering Reports,* vol. 8, no. 7, pp. 78.

[21] Kosim, S.A., Bagus, J.S., Deka, J.A., Achmad, R.M., I Nyoman Gede Artadana,Mahaputra Wardhiana & Royyana, M.I. 2026, "Machine Learning-Driven Resource Provisioning in Modern Cloud Environments: A Taxonomic Survey", *International Journal of Advanced Computer Science and Applications,* vol. 17, no. 4, pp. 14.

[22] Lakshmi, G. & Swetha, P. 2026, "NeuroSymbolicPhishDefend for adaptive multimodal phishing detection against evolving artificial intelligence driven threats", *Scientific Reports (Nature Publisher Group),* vol. 16, no. 1, pp. 28177.

[23] Lu, Y. 2026, "Research and implementation of intelligent clothing personalized customization system based on deep learning", *Scientific Reports (Nature Publisher Group),* vol. 16, no. 1, pp. 12080.

[24] 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.

[25] Marimuthu, G. 2026, "Modernizing Foreign Exchange Cross-Currency Trading: Challenges of Outdated Banking Infrastructure and the Path Toward Artificial Intelligence-Driven Low-Latency Technology", *Journal of International Crisis and Risk Communication Research,* vol. 9, no. 1, pp. 45-52.

[26] Michailidis, E.T. & Karanasiou, I.S. 2026, "A Review of AI-Enabled UAV-Based Systems for Defense Applications", *Drones,* vol. 10, no. 8, pp. 602.

[27] Mohanad, A. & Jaradat Ahmad, M. 2026, "A Cross-Layer Review of Intelligent, Secure, and Privacy-Preserving Internet of Vehicles", *Machine Learning and Knowledge Extraction,* vol. 8, no. 9, pp. 277.

[28] Olagundoye, O.O., Olusola, B., Chukwuebuka, J.E., Oluwatoyosi, B., Ting, N. & Onyango, V. 2026, "A Review of Artificial Intelligence Techniques for Low-Carbon Energy Integration and Optimization in Smart Grids and Smart Homes", *Processes,* vol. 14, no. 3, pp. 464.

[29] Ortega-Del-Rosario Maria De, L.A., Caballero, R., Domínguez Max, A.M., Romas, L., Noguera, J.C., Jaén-Ortega, A.A. & Castaño, C. 2025, "AI-Enhanced Manufacturing in Latin America: Opportunities, Challenges, Applications, and Regulatory Policy Frameworks for Intelligent Production Systems", *Applied Sciences,* vol. 15, no. 20, pp. 11056.

[30] Reshman, J., Manoochehr, R. & Wamadeva, B. 2026, "Real-Time Synchronisation in Low-Power Wireless Sensor Networks: From Industry to Healthcare", *Technologies,* vol. 14, no. 7, pp. 394.