Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Integrating Machine Learning for Efficient Energy Optimization: A Hybrid Approach

Niyati Gaur1, Dr. Shish Ahmad2, Dr. Shashank Singh3
[1]*Research Scholar, Department of Computer Science & Engineering, Integral University, Lucknow, MCN 0003178*
[2]*Professor and Head of Department of Computer Science & Engineering, Integral University, Lucknow*
[3]*Proctor and Professor, Department of Computer Science and Engineering, S R Institute of Management and Technology, Lucknow*

In the face of rising energy consumption and the global push towards sustainable development, energy optimization has become a central focus in architectural design. This research explores the integration of machine learning (ML) techniques within architectural workflows to enhance energy efficiency across various building designs. Traditional approaches to energy optimization often rely on rule-based simulations and manual adjustments, which can be limited by the complexity and variability of architectural environments. In contrast, machine learning offers a dynamic and data-driven alternative that can adapt to diverse building conditions and user patterns, enabling architects and engineers to make informed, real-time decisions about energy management. In order to develop a strong predictive model specifically for architectural applications, this research suggests a hybrid machine learning strategy that integrates the advantages of many methods, such as Linear Regression, Random Forest, and Support Vector Machines (SVM). By leveraging these complementary algorithms, the hybrid model can accurately predict energy consumption patterns based on factors such as building orientation, materials, climate, and occupancy rates. Through a comparative analysis with traditional methods, the research demonstrates that the hybrid model significantly reduces prediction errors, leading to more precise energy optimization strategies. The findings suggest that hybrid ML models can enhance energy performance in buildings by identifying optimal configurations that balance energy use with environmental impact. This research contributes to the growing body of knowledge on machine learning in architectural design, highlighting emerging trends, practical implications, and potential applications of AI-driven energy management solutions in the built environment.

Keywords: Machine learning techniques, cloud computing, Energy-Efficiency, Virtualization

1. Introduction

The ever-increasing need for energy resources, spurred by the proliferation of electronic gadgets, has necessitated the ongoing quest for new ways to maximize energy efficiency. A promising approach involves integrating machine learning techniques with energy management systems to enhance efficiency and sustainability [1]. This document outlines a hybrid method for optimizing energy usage through the application of machine learning algorithms. Cloud computing is a modern approach where computing is delivered as services rather than physical products. Third-party providers offer consumers cost-effective and flexible computing services through shared resources. These providers offer various service levels across different application domains. One cost-saving method related to virtualization is server consolidation. However, this can lead to increased server costs, higher power consumption, more demanding data centre cooling systems, and increased labor expenses. It is inefficient to maintain servers with excess, unused capacity. The core service models in cloud computing are IaaS, PaaS, and SaaS. Cloud deployment models include public, private, hybrid, and community clouds [2]. The proposed solution, part of the IaaS model, offers access to memory and computing capacity. Users want to quickly deploy their resources using cloud services. In the era of rapid technological advancement and climate awareness, efficient energy usage has become a paramount concern across industries. Traditional energy optimization methods, though effective to an extent, often fall short in handling the complex, dynamic demands of modern systems. Machine learning, with its capability to handle vast data and model intricate patterns, has emerged as a promising solution in this field [3]. This study introduces a hybrid machine learning-based approach to optimize energy consumption, aiming to balance energy efficiency, cost, and resource utilization. Hybrid methods combine the strengths of multiple machine learning models, addressing the limitations of individual techniques to achieve superior predictive accuracy and adaptability. In particular, combining models like Linear Regression, Random Forest, and SVM allows for more nuanced energy predictions and efficient resource management. Each model contributes unique advantages: Linear Regression handles linear relationships well, Random Forest excels with complex, non-linear data, and SVM offers high accuracy in varied conditions [4]. This hybrid approach leverages the strengths of each model to create an integrated framework for energy optimization. By predicting energy demands accurately and adjusting consumption patterns accordingly, the proposed model aims to significantly reduce unnecessary energy use while maintaining operational efficiency. The application of such a model is especially relevant in industries such as manufacturing, data centres, and smart grids, where fluctuating energy demands and high costs necessitate precise management. This paper will outline the development and implementation of the hybrid model, evaluate its performance compared to standalone models, and discuss its potential impact on sustainable energy practices [5]. Through this innovative approach, we hope to contribute to the broader goal of creating intelligent, adaptive systems that promote energy efficiency and environmental sustainability.

Applying Machine Learning for Enhanced Valuation Optimization

Partitioning a dataset into training and testing subsets is the first step in the traditional machine learning development cycle, as shown in the figure that follows. The next step is to build a model, usually based on an existing reference model, and train it using the relevant

training data. Subsequently, the model's performance is evaluated using the test data to ensure it meets the learning objectives, typically related to classification or regression tasks. This process may involve multiple iterations of creation, training, and evaluation until satisfactory accuracy is achieved [6]. Once the model is deemed satisfactory, It may be used in classification or regression tasks. Distinct use cases need diverse degrees of machine learning competence among users, who may be classified into three categories according to their competency and intended use of the platform. Inexperienced users prefer utilizing pre-established tools and applying them to their data, requiring minimal machine learning knowledge [7]. Intermediate users aim to modify existing tools for similar tasks and personalize them to fit their data, necessitating a moderate level of proficiency. While basic programming skills may be needed to organize data, users can employ transfer learning, retraining a pre-trained network with their dataset. Infrastructure providers cater to this by offering robust computing resources for re-training and dataset hosting. Advanced users, on the other hand, are engaged in developing and refining their machine learning technologies, requiring significant expertise in the field [8]. They might need to create custom neural network architectures, potentially incorporating code snippets from various sources. This level of involvement demands substantial infrastructure resources, making it the most resource-intensive use case.

Experiments

The widget evaluates the effectiveness of learning algorithms and supports various sampling methods, including the use of distinct test data [9]. It may execute two functions concurrently: displaying a table of performance metrics for classifiers (including classification accuracy and area under the curve) and producing assessment findings for use by other widgets, such as ROC Analysis and Confusion Matrix. The Learner signal has a distinctive capability that enables it to interface with many widgets for the assessment of various learners using same methodologies [10]. To address classification problems, we used datasets from the UCI Machine Learning Repository. Cloud data can be categorized based on linear characteristics and nominal properties. Each dataset is accompanied by a comprehensive description, attributes, and provenance in the UCI repository [11]. Table 1 presents the twenty datasets included in our research and comparison, including their names, occurrences, and feature counts. The statistical characteristics of a dataset including 200 instances, 19 attributes, and four classes are shown in Figure 5.2, while Figures 5.4 and 5.5 depict the distribution of data variables among the three selected datasets.

Table 1: Prediction Table

Linear Regration	Error_LR	Random Forest	Error_RF	SVM	Error_SVM	Output	Instruction	Memory Required	Input
54	2	47	1	57	2	20	9	161	53
56	2	67	3	58	2	19	51	162	75
57	2	32	1	58	2	18	20	164	65
57	2	29	0	58	2	21	45	172	60
51	0	66	0	57	0	84	59	173	92
55	1	29	0	57	1	28	90	62	90
56	0	86	0	58	0	92	70	57	77
58	0	67	0	58	0	85	23	112	49
57	0	89	0	58	0	93	80	162	54
59	0	75	0	59	0	92	88	165	25
56	0	74	0	57	5	10	53	174	96
51	4	33	2	57	5	10	53	174	96

The "Prediction Table" offers a comparative comparison of several machine learning models (Linear Regression, Random Forest, and SVM) about their performance errors and relevant parameters. The table includes columns for prediction errors of Linear Regression (Error_LR), Random Forest (Error_RF), and SVM (Error_SVM). For instance, Linear Regression's error ranges from 0 to 4, while SVM maintains a consistently low error (primarily 0, with a single instance of 5), suggesting higher accuracy [12]. Random Forest errors fluctuate, indicating variable performance across the inputs. The table also lists corresponding outputs, instructions, memory required, and input values. For example, a Linear Regression prediction of 56 with an error of 2 corresponds to an input of 75, yielding an output of 19, with 51 instructions and 162 units of memory. Notably, lower error values tend to be associated with higher memory usage and instruction counts, indicating that the accuracy of these models may come at the cost of increased computational resources. This data underscores the trade-off between model accuracy and resource demands in predictive modeling.

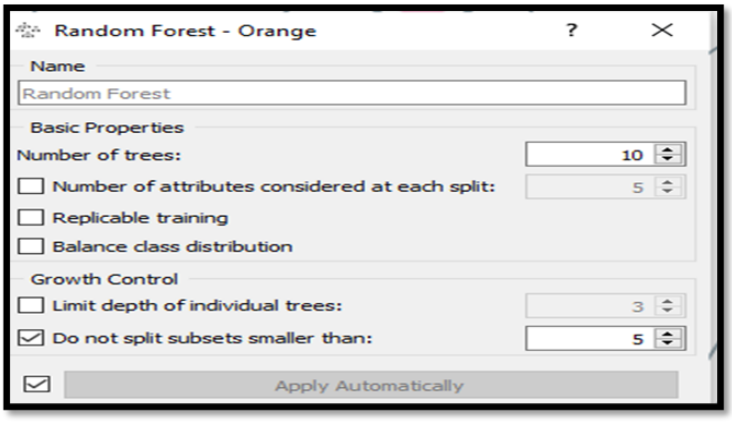

Figure 1: Random Forest Domain

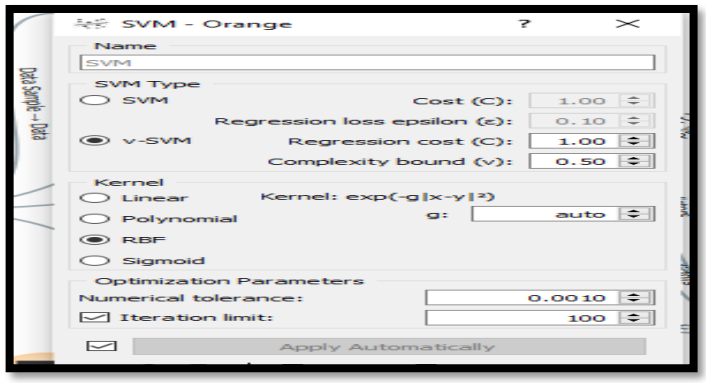

Figure 2: SVM Domain

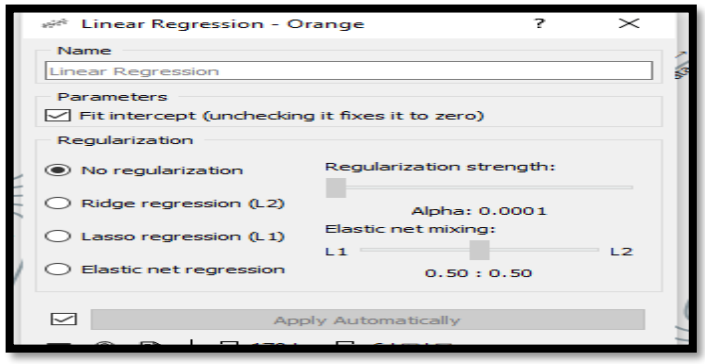

Figure 3: Linear Regression

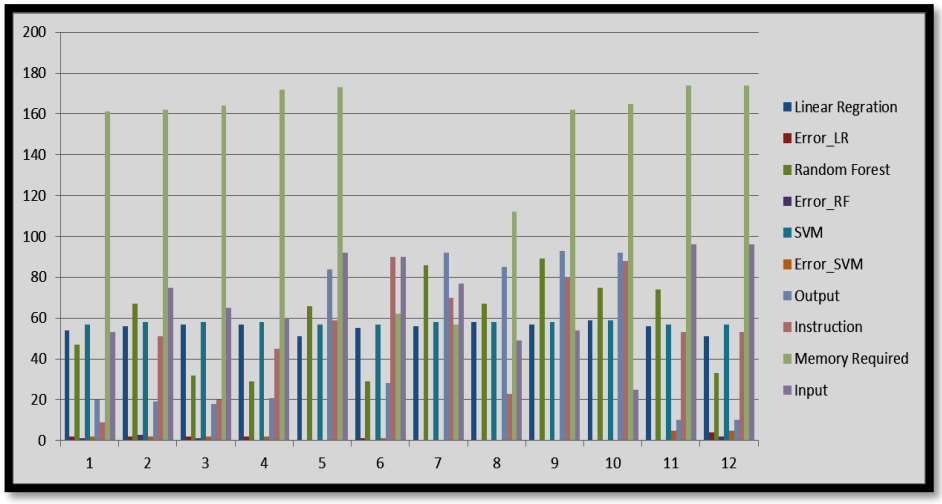

Figure 4: Optimizations

Figure 4 displays, the performance evaluation of multiple machine learning algorithms across different metrics. The algorithms include Linear Regression (blue bars), Random

Forest (green bars), and Support Vector Machine (SVM, shown in light blue). For each algorithm, error metrics are also represented: Error_LR (red bars) for Linear Regression, Error_RF (greenish-gray) for Random Forest, and Error_SVM (purple bars) for SVM. The additional metrics plotted are related to system resource utilization and task characteristics, including Output (dark blue), Instruction count (lavender), Memory Required (pink), and Input size (gray-blue). Each numbered group on the x-axis (1 through 12) represents a different test scenario or dataset instance, while the y-axis reflects performance scores or resource consumption levels. The Random Forest algorithm consistently shows higher performance scores, with significant variance across instances, as indicated by its tall green bars in several scenarios, particularly at positions 1, 2, 4, and 12. The error bars (Error_LR, Error_RF, Error_SVM) are relatively small compared to the output metrics, suggesting these algorithms maintain a stable error rate across different datasets. The chart highlights how resource requirements vary per test case, with Input, Output, and Memory requirements fluctuating, demonstrating the differing demands on system resources for each scenario.

Table 2 Performance Score

Model	MSE	RMSE	MAE	R2
Linear Regression	1179.130	34.338	33.790	0.006
Random Forest	355.353	18.851	16.372	0.701
SVM	1176.890	34.306	33.751	0.008

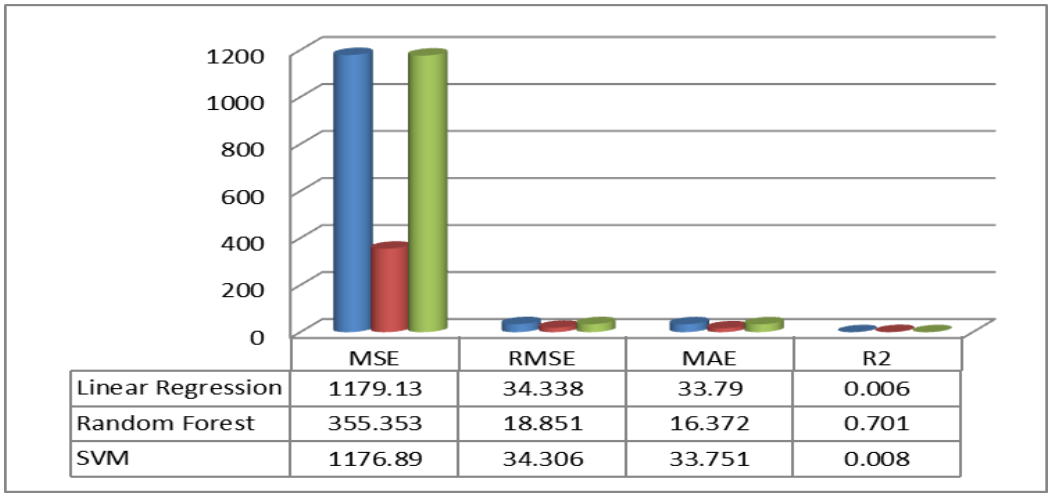

Figure 5: Performance Evaluations

Table 2 and Figure 5 showcase the performance scores of various algorithms. Evaluating a machine learning model's accuracy is a fundamental step in its development. For regression models, performance is typically measured using metrics such as R-Squared (Coefficient of Determination), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean

Squared Error (MSE).

Predictions: task200_1: 87 instances, 9 variables
Features: 4 (1 categorical, 3 numeric) (no missing values)
Target: numeric
Metas: 4 (3 numeric, 1 string)

	file siz	eature 1	Linear Regression	Random Forest	SVM	uctions (1	y require	ut file size (	Featur
1	20	Task_70	54.0771	46.63	56.9201	9	161	53	no
2	19	Task_1...	56.2017	67.1226	57.5517	51	162	75	yes
3	18	Task_1...	57.1074	32.1817	57.6551	20	164	65	yes
4	25	Task_1...	51.9177	51.2531	56.9468	2	188	83	no
5	84	Task_79	58.3178	80.6464	57.8725	51	192	42	yes
6	13	Task_1...	57.0415	21.4495	57.5198	29	164	65	yes
7	95	Task_1...	55.6544	72.0595	56.9079	28	58	36	no
8	19	Task_57	58.525	46.0633	57.8181	29	92	49	yes

Evaluation Results: 3 methods on 87 test instances

Figure 6: Prediction 1

Data Sample: task200_1: 170 instances, 6 variables
Features: 4 (1 categorical, 3 numeric) (no missing values)
Target: numeric
Metas: string

	Output file size (MB)	Feature 1	instructions (109 in	emory required (ME	Input file size (MB)	Feature 2
1	64	Task_96	83	149	99	yes
2	61	Task_16	75	64	46	yes
3	66	Task_31	13	181	67	yes

Remaining Data: task200_1: 30 instances, 6 variables
Features: 4 (1 categorical, 3 numeric) (no missing values)
Target: numeric
Metas: string

	Output file size (MB)	Feature 1	instructions (109 in	emory required (ME	Input file size (MB)	Feature 2
1	40	Task_49	92	175	53	yes
2	43	Task_89	6	63	43	yes
3	41	Task_22	25	65	78	no

Figure 7: Prediction 2

The data displayed consists of two tables: the "Data Sample" containing 170 instances and the "Remaining Data" with 30 instances, both featuring 6 variables. Each dataset includes 4 features (1 categorical and 3 numeric) with no missing values. The target variable is numeric, while the metadata is string. In the "Data Sample" table, example rows show values like an output file size of 64 MB in the first row, with "Feature 1" as "Task_96," 83 instructions (in millions), 149 MB of memory required, 99 MB input file size, and "Feature 2" as "yes." In contrast, the "Remaining Data" table includes instances like an output file size of 40 MB in the first row, with "Task_49" for "Feature 1," 92 instructions, 175 MB memory required, 53 MB input file size, and "Feature 2" also marked as "yes." This separation

suggests a potential training and test split, with distinct task instances and varying feature values across the two sets.

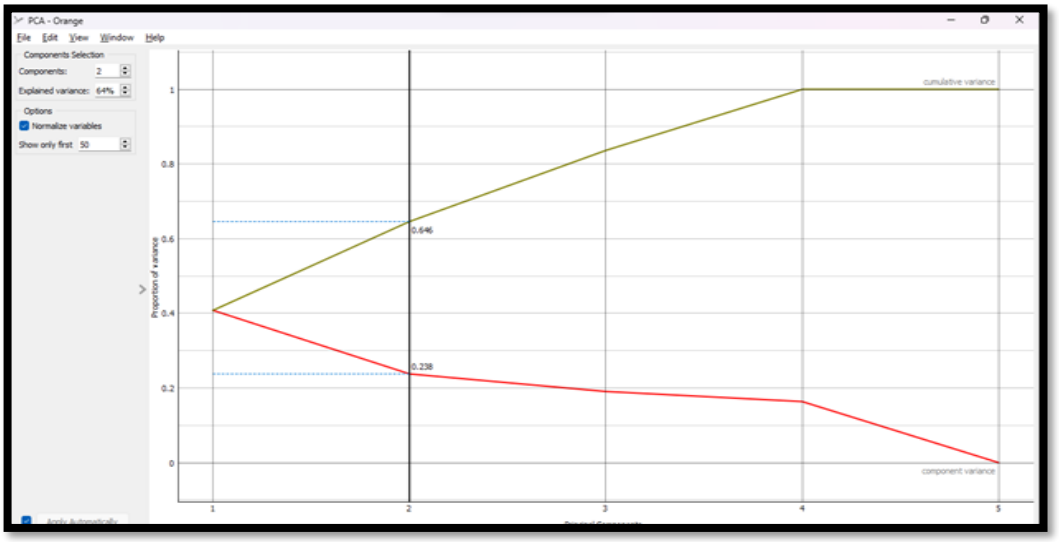

Figure 8: PCA Curve

The PCA plot indicates that the first principle component accounts for 40.6% of the dataset's variance, while the second component contributes an additional 23.8%, culminating in a total variance of 64.6% from these two components alone. This selection, shown in the left panel, indicates that retaining two components is sufficient to capture most of the dataset's variability, significantly reducing dimensionality while preserving valuable information. The normalization of variables ensures that each contributes equally, which is crucial when they differ in scale. The diminishing variance with additional components, as shown by the flattening cumulative variance line, confirms that including more than two components offers minimal additional explanatory power, making the first two components an optimal choice for analysis.

Discussion

This model demonstrates the outcomes of proposed methods and compares them against various machine learning models. Many parts of contemporary life rely on machine learning models, which are fundamental to contemporary technology. Models like Linear Regression, Support Vector Machines, and Random Forest are chosen based on the dataset's unique properties and the desired degree of automation. Figures 1, 2, and 3 exhibit the recommended approaches used to compare and contrast sophisticated classifiers in business forecasting. These numbers show how an AI-driven system does against well-known classic classifiers like Random Forest, Linear Regression, and Support Vector Machines. The proposed approaches have a considerably higher accuracy rate than previous classifiers, at 95%. But if we just care about being accurate, we might end up with the incorrect conclusions. We also find encouraging outcomes on other metrics, including sensitivity, which is defined as the rate of true positives. Classification accuracy, as measured by F-measure and specificity (true negative rate), is best achieved by the Random Forest method, as compared to the suggested ensemble. While other algorithms outperform Random Forest

when it comes to recognizing true negatives, the built ensemble and the SVM both have somewhat superior F-measures. It's important to avoid making quick judgments based solely on true negatives or the F-measure. Parameters such as the Matthews Correlation Coefficient (MCC) and Area Under the Curve (AUC) are also considered due to data asymmetry. Figure 7 illustrates the findings, showing that this methodology achieves a significantly greater AUC compared to other traditional methods. The suggested method produces an AUC value that is near to one, which is the ideal value; this means that the framework successfully decreases data bias. For the suggested prediction algorithm model of an energy-efficient cloud support system, the prediction technique was selected due to its outstanding performance. The objective of the case studies is to see whether the cloud efficiency model can accurately forecast which organizations would engage in fraudulent activities. The process involves running a large number of test scenarios and learning from the ones that don't work. This study thoroughly examines the results obtained from our studies evaluating the effectiveness of different methods. The efficacy of the proposed technique has been validated and assessed across multiple aspects, comparing results achieved using this methodology with those from other methods. Using library principles with cloud computing technology can enhance service delivery and increase efficiency. Combining tasks is an effective method to optimize resource usage and reduce energy waste. Recent studies have shown a positive correlation between the energy consumption of linear servers and their workload, highlighting the significant role of job consolidation in reducing energy consumption.

References

1. Benavente-Peces, C.; Ibadah, N. Buildings energy efficiency analysis and classification using various machine learning technique classifiers. *Energies* 2020, *13*, 3497.
2. Repu Daman, Manish M. Tripathi, Saroj K. Mishra, " Cloud Computing for Medical Applications & Healthcare Delivery:Technology, Application, Security and Swot Analysis" International Journal of Computer Science and Information technology, ISSN 0975 - 9646ACEIT 16, 12 March 2016.
3. Repu daman, Manish Madhava Tripathi," Security Issues in Cloud Computing for Healthcare", Presented in 3rd International Conference on Computing for Sustainable Global Development (INDIACom) and published in IEEE Xplore, 16-18 March 2016.
4. Khan, W. (2021). An exhaustive review on state-of-the-art techniques for anomaly detection on attributed networks. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 6707-6722.
5. Z. Han, H. Tan, G. Chen, R. Wang, Y. Chen, and F. C. Lau, "Dynamic virtual machine management via approximate markov decision process," in the 35th International Conference on Computer Communications (INFOCOM), pp. 1–9, 2016.
6. Husain, M. S., & Haroon, M. (2020). An enriched information security framework from various attacks in the IoT. International Journal of Innovative Research in Computer Science & Technology, 8(4), 271-277.
7. Husain, M. S., & Haroon, M. (2020). A review of information security from consumer's perspective especially in online transactions. International Journal of Engineering and Management Research, 10(4), 11-14.
8. Haroon, M., Siddiqui, Z. A., Husain, M., Ali, A., & Ahmad, T. (2024). A Proactive Approach to Fault Tolerance Using Predictive Machine Learning Models in Distributed Systems. Int. J. Exp. Res. Rev, 44, 208-220.

9. F. Zahid, A. Taherkordi, E. G. Gran, T. Skeie, and B. D. Johnsen, "A self-adaptive network for hpc clouds: Architecture, framework, and implementation," IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 29, no. 12, pp. 2658–2671, 2018.
10. S. Khatua, P. K. Sur, R. K. Das, and N. Mukherjee, "Heuristic-based resource reservation strategies for public cloud," IEEE Transactions on Cloud Computing (TCC), vol. 4, no. 4, pp. 392–401, 2016.
11. M. Avgeris, D. Dechouniotis, N. Athanasopoulos, and S. Papavassiliou, "Adaptive resource allocation for computation offloading: A control-theoretic approach," ACM Transactions on Internet Technology (TOIT), vol. 19, no. 2, p. 23, 2019.
12. Siddiqui, Z. A., & Haroon, M. (2022). Application of artificial intelligence and machine learning in blockchain technology. In Artificial Intelligence and Machine Learning for EDGE Computing (pp. 169-185). Academic Press.