SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Cloud Computing Systems: Performance Trade-Offs In Scalable Architectures

Aarsi Kumari[1], Tamilselvan Thangavel[2], Ebin Horrison S[3], Sathya Arthi R[4], S. Balaji[5], Samundeeswari K[6], Yusupova Dildora Uktamovna[7]

[1] Assistant Professor, Department of Computer Science & IT, Arka Jain University, Jamshedpur, Jharkhand, India. Email: aarsi.k@arkajainuniversity.ac.in, ORCID: 0009-0008-5355-234X
[2] Assistant Professor, Department of Computer Science and Engineering, Presidency University, Bangalore, Karnataka, India. Email: tamilselvan.t@presidencyuniversity.in, ORCID: 0009-0004-4372-7434
[3] Professor, Department of Architecture, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India. Email: ebinhorrison.arch@sathyabama.ac.in, ORCID: 0000-0002-6603-1024
[4] Assistant Professor, Department of Management Studies, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: sathyaarmba@maher.ac.in
[5] Professor, Department of CSE, Panimalar Engineering College, Chennai, Tamil Nadu, India. Email: balajiit@gmail.com, ORCID: 0000-0003-3208-1637
[6] Assistant Professor, Department of Commerce, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: kscommerce@maher.ac.in
[7] Teacher, Department of Physiology, Faculty of Administration and Management, Samarkand State Medical University, Samarkand, Uzbekistan. Email: usupovadildora8@gmail.com, ORCID: 0009-0008-1455-8810

Abstract
Cloud computing systems play a critical role in supporting scalable applications in domains such as e-commerce, where highly dynamic user demand requires efficient resource allocation. Auto-scaling mechanisms enable cloud platforms to dynamically adjust resources in response to workload variations while maintaining Service Level Agreement (SLA) requirements. However, achieving an optimal balance between system performance and operational cost remains challenging due to complex interactions among workload patterns, resource configurations, and scaling strategies. This research proposes an optimization based on a Bear Smell Search Feed Forward Neural Network (BSS-FFNN) model, where the BSS algorithm is employed to explore the search space and identify optimal resource configuration parameters, while the FFNN is utilized to learn workload patterns and predict system performance for informed decision-making in resource management. An e-commerce workload dataset comprising 11,000 samples is utilized for evaluation. Data preprocessing includes Min-Max normalization and outlier removal based Interquartile Range (IQR) method to ensure data consistency. Feature extraction is performed using Principal Component Analysis (PCA) to identify significant workload characteristics. The proposed method identifies optimal configurations that balance response time, throughput, and cost while ensuring SLA compliance. Experimental results demonstrate a response time of 36ms, a throughput of 600 req/ms, Central processing unit (CPU) utilization of 49 %, and memory utilization of 35 %, and the implementation is carried out in Python using deep learning (DL) libraries. In conclusion, the BSS-FFNN-based method provides a scalable and efficient solution for optimizing performance–cost trade-offs in cloud-based e-commerce environments.

Keywords: Cloud Computing, Auto-Scaling, E-commerce, Performance Optimization, Cost Efficiency.

1. Introduction

Today, cloud computing and E-Commerce systems have become crucial components of modern digital infrastructure, offering scalability of resources access, effective handling of workloads and connecting businesses on a global level. Workload balancing, multi-cloud integration, and system optimization in distributed environments pose challenges in the form of latency, inefficient resource usage, and downtime that may have a significant impact on system efficiency and customer satisfaction [1,2]. Furthermore, with the emergence of cross-border commerce, system complexity has been further complicated due to heterogeneity

of data, global transactions, and logistics challenges. To maintain operations and enhance service delivery in the face of growing complexity, state-of-the-art technologies need to be applied [3]. In particular, current developments seek to utilize AI/ML/DL-based approaches in order to solve various problems within cloud and E-Commerce environments [4,5]. Moreover, AI-based methods provide adaptive autoscaling capabilities, whereas DL algorithms improve recommendation engines, forecast demand, deliver personalization services, and facilitate decision intelligence by managing voluminous and multidimensional e-commerce data efficiently [6]. All of these methodologies altogether make cloud and e-commerce ecosystems smarter.

However, despite the above improvements, several challenges still exist when implementing highly efficient and reliable e-commerce cloud systems. The conventional rule-based and reactive approaches have been proven inefficient with regard to allocating resources efficiently, latency problems, and increased costs due to their inability to handle unexpected workloads [7]. In addition, multi-cloud systems pose challenges in terms of synchronization, optimizing performance, and cost optimization, whereas ML/DL systems have challenges of interpretability, scaling up, and privacy issues [8]. It is therefore important to develop more efficient, robust, and intelligent ways of managing such systems.

Research aim: The research's main goal is to enhance the management of workloads within cloud computing and e-commerce through addressing issues of resource allocation, scalability, latency, and efficiency. The goal of the research is to create a scalable and efficient approach for e-commerce through the application of AI/ML/DL approaches.

Research organization: The background of cloud computing and e-commerce, with respect to workload balancing and multi-cloud management, is highlighted in section 1. Section 2 provides information on previous research done in the area, employing AI, ML, and DL in workload management, automatic scaling, and efficient resource utilization. The proposed methods utilizing ML and cloud-native technologies will be summed up in section 3. Different types of evaluation experiments carried out on the BSS-FFNN model have been compared to different baseline models in section 4. Section 5 will offer the research's conclusion.

2. Related work

Recent developments in cloud computing, e-commerce systems, and AI-based optimizations are discussed below in Table 1. All the research focuses on the application of several essential techniques like ML models, streaming systems, optimization methods, and cloud architectures. Almost all the research tries to address the issue of scalability, efficiency, cost-effectiveness, and SLA compliances. Generally speaking, findings reveal enhanced efficiency of systems, but often, some drawbacks exist like complex system architecture, hardware dependencies, and inability to deal with huge amounts of data.

Table 1: Recent research on cloud computing and e-commerce systems

Ref	Research Aim	Technique Used	Experimental Results	Limitation
[9]	Enable efficient adaptation and management of e-commerce Internet of Things cloud systems using automated monitoring and control	Policy-driven framework with metric-based monitoring	Improved efficiency, reliability, and SLA alignment	Handling highly volatile large-scale IoT data streams
[10]	Examine how digitalization has affected e-commerce logistics and the function of smart logistics.	Systematic Literature Review + network analysis (288 Scopus articles)	Identified Information and Communication Technology and Smart Logistics clusters are improving tracking, responsiveness	Limited computer vision use, product inspection gaps, and data volume issues
[11]	Optimize cloud auto-scaling for cost-efficient SLA-compliant configurations	Stochastic Petri Nets + Greedy Randomized Adaptive Search Procedure	Cost-efficient SLA-compliant configurations using 18k dataset	Complex modeling of dynamic heterogeneous cloud systems
[12]	Evaluate small open-weight Large language models for	Large Language Model Meta AI 3.2 (1B) + Quantized Low-Rank	98.8% accuracy; Generative Pre-trained Transformer	Hardware-dependent trade-offs,

	multilingual e-commerce intent recognition vs large models	Adaptation (QLoRA) + GPTQ + Generated Unified Format (GGUF) quantization	Quantization (GPTQ) decreases Video Random Access Memory (VRAM) 41%; GGUF 4.3× CPU speedup	quantization inefficiency
[13]	Create scalable data integration for e-commerce platforms with multiple retailers.	Schema matching + ontology mapping + ML-based entity resolution + Spark + Kubernetes	Improved catalog accuracy & operational efficiency	Heterogeneous data integration + real-time consistency issues
[14]	Design secure and scalable cloud-based e-commerce platform	Elastic cloud + multi-layer security + performance analysis	Improved performance & security under dynamic load	Cloud security issues + workload variability
[15]	Use real-time streaming architecture to update batch pipelines.	Apache Kafka + Spark Structured Streaming + Google BigQuery	Reduced latency, higher scalability vs batch systems	Higher complexity + operational cost
[16]	Build hyper-scalable AI framework for e-commerce personalization	Microservices + containerization + load balancing + hybrid ML models	High accuracy, strong scalability, slight latency under peak load	Latency increases under extreme traffic

Research gap: Current cloud and e-commerce methods have problems with high latency, adapting to changing workloads, resource allocation, and scaling up in multi-cloud setups. Model BSS-FFNN addresses such problems by leveraging AI/ML-based intelligent prediction and optimization and real-time workload management. This greatly improves efficiency and scalability and ensures better SLA compliance.

3. Methodology

This study proposes a methodology to optimize cloud utilization for e-commerce application with BSS-FFNN. First, the Min-Max normalization and IQR were used to preprocess the data, then PCA was utilized for feature selection. For making appropriate decisions, BSS algorithm and FFNN were implemented for identifying workload behavior prediction in cloud. The entire process was implemented in Python using some DL libraries. Figure 1 illustrates how this suggested method for resource, cost, and performance optimization within SLA was validated using 11,000 sample data.

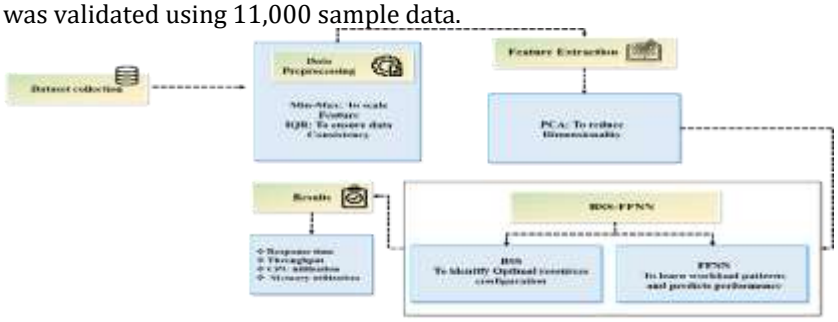

Figure 1: Methodology of the proposed model

3.1 Dataset description

The research uses an open-source dataset (https://www.kaggle.com/datasets/colabsss/cloud-performance-dataset) from Kaggle with 11,000 records. It covers cloud computing performance and costs in China's e-commerce sector. Dataset comprises of workload information, cloud configuration details, resource consumption information, performance statistics, availability information, cost statistics, SLA information, and ultimate performance trade-offs. Therefore, the dataset will be perfect to research on the management of high-demanding online applications such as retail, live-commerce, food-delivery, cross-border shopping, and digital payment solutions in the China region. The dataset is divided into an 80% training set and a 20% testing set for model development.

3.2 Preprocessing of the dataset

The Min-Max normalization approach is an effective way of scaling the cloud computing data during the preprocessing phase in such a way that all the values fall within the same range of either [0,1] or another

specific range of values. The other approach that can be used to preprocess cloud computing data involves the use of the IQR approach to detect and eliminate outliers from the dataset.

3.2.1 IQR for data cleaning

By calculating the difference between the first and third quartiles (Q1 and Q3) and choosing any value that falls beyond the ranges given by Q1 − 1.5×IQR to Q3 + 1.5×IQR, the IQR methodology, a technique used in cloud computing, finds and eliminates outlying data points. The primary goal of using the IQR approach in cloud computing is to enhance data quality by getting rid of any outlying data point.

$f(x) \leftarrow d_i = 0, x_i \leq 0$ (1)

$IQR = q_3 - q_1$ (2)

In Equation (1), $f(x)$ represents the transformed output function, d_i denotes the assigned decision or processed value for the i^{th} data point, and x_i is the original input feature value. The condition $x_i \leq 0$ indicates that the transformation is applied when the input value is less than or equal to zero, which is often used in preprocessing or activation-based filtering steps to control invalid or non-positive inputs. In equation (2), IQR stands for Interquartile Range, which measures statistical dispersion in the dataset, the first quartile is represented by q_1, and the third quartile by q_3.This difference $q_3 - q_1$ captures the spread of the middle 50% of the data and is commonly used for detecting and removing outliers.

3.2.2 Min-Max normalization

One popular technique in cloud computing is the min-max normalization method, which scales numerical features to a range of values. The min-max normalization technique focuses on maintaining the relations among different data points while scaling their features.

$V_{normalized} = \frac{V - V_{min}}{V - V_{min}}$ (3)

The min-max normalization equation is shown in equation (3), where $V_{normalized}$ is the transformed feature's scaled value. V_{max} and V are the maximum and minimum values of the features, while V is the initial input feature value.

3.3 Dimensionality reduction using PCA

PCA is one method that allows high-dimensional data to be transformed into low-dimensional data without sacrificing any information. Its main objective in cloud computing applications is to reduce feature complexity, remove redundancy, and highlight the most significant patterns in the dataset. This helps improve computational efficiency and enhances the performance of ML models deployed in cloud computing systems.

$N^{m*m} = n_{j,i}, (n_{j,i} = Cov(B_j, B_i))$ (4)

$Cov(B_j, B_i) = \frac{\sum_{l=1}^{m}(B_{jl} - B_j')(B_{il} - B_i')}{m-1}$ (5)

$Nv = \delta v$ (6)

From the above equation (4-6), N^{m*m} represents the size of the covariance matrix, where m is the total number of samples. $n_{j,i}$ is the Covariance value between feature B_j and feature B_i, and B_j and B_i are individual features. B_{jl} and $B_{il}$are the value of feature B_j and B_i at sample l and l is the index of each sample observation. B_j' and B_i' are the mean values of feature B_j and B_i. Cov denotes the covariance between two variables, B_j and B_i,$\sum_{l=1}^{m}$ represents the summation over all m samples in the dataset used to compute covariance. N is the covariance matrix of the dataset, where v is the original value, and δ is a scalar value.

3.4 FFNN- BSS Methods for Cloud Resource Optimization

An FFNN is a DL model used in cloud computing that forecasts system performance and captures complex nonlinear data correlations in cloud environments. It does this by sending information directly from the input to the output layers. The model's aim is to support smart cloud computing resource management decisions by accurately mapping workload patterns. The BSS algorithm is a nature-inspired optimization technique that identifies the best cloud resource configuration settings by searching the parameter space. The above procedure is a balance between exploration and exploitation that tries to improve global optimization for the cloud computing environment. This is accomplished through efficient trade-offs between performance and cost.

3.4.1 FFNN for Cloud Workload Prediction

A FFNN is a form of supervised learning algorithm used in cloud computing, in which information moves linearly through input and hidden layers and ends up at the output layer, allowing the model to effectively learn nonlinear mappings of information within cloud computing datasets. The ultimate goal of this algorithm is to make accurate predictions on system performance based on cloud computing workload patterns.

$$n = 2m + 1 \quad (7)$$

$$a_j^{out} = f\left(\sum_{=1}^{m} w_{ij} x_i + b_j\right) \quad (8)$$

$$f(u) = \frac{1}{1+e^{-u}} \quad (9)$$

$$o_q^{out} = f\left(\sum_{j=1}^{n} w'_{jq}\ a_j + b'_q\right) \quad (10)$$

Equation (7) establishes the size of the hidden layer depending on the input dimension, where n is the total number of neurons in the hidden layer and m is the number of input characteristics or input neurons. The output of the j^{th}hidden neuron is denoted by a_j^{out} in equation (8). The activation function is $f(\cdot)$, the weight between the i^{th} input neuron and the j^{th} hidden neuron is w_{ij}, thei^{th}input feature is x_i, the bias term for thej^{th}hidden neuron is b_j, and the summing process is represented by the notation $\sum_{=1}^{m}$. Equation (9) is the sigmoid activation function, where u represents the weighted sum input to the neuron and maps values into the range (0,1), and the term e^{-u} represents the exponential function where e is Euler's constant and $-u$ is the negative value of the input u. The output of the q^{th} output neuron is denoted by o_q^{out}in equation (10), the weight between thej^{th} hidden neuron and the q^{th} output neuron is denoted by w'_{jq}, the output from the j^{th}hidden neuron is denoted by a_j, the bias term for the output neuron is denoted by b'_q, and the sum of the contributions from all n hidden neurons in the neural network is represented by $\sum_{j=1}^{n}$.

3.4.2 BSS for search space

The term BSS refers to a cloud computing optimization technique that draws inspiration from nature. In order to identify the best cloud computing solution, the BSS algorithm employs an analogy of bear foraging behavior, which entails exploration and exploitation. Important factors including population size, iteration count, and control parameters α and β are adjusted during the tuning phase to regulate the process's pace of convergence and improve search accuracy, preventing local optima and producing high-quality solutions.

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \quad (11)$$

$$O_i = f(X_i) \quad (12)$$

$$X_i^{t+1} = X_i^t + r \cdot (X_{best}^t - X_i^t) \quad (13)$$

$$X_i^{t+1} = \begin{cases} X_i^{t+1}, if\ f(X_i^{t+1}) < f(X_i^t) \\ X_i^t \qquad\quad , otherwise \end{cases} \quad (14)$$

The position vector of thei^{th} solution in a d -dimensional search space is represented by equation (11), where each x_{ij} represents the value of the i^{th} solution in the d^{th} dimension of the search space and is a decision variable corresponding to the j^{th} parameter. The fitness value of solution X_i is represented by O_i in Equation (12), where $f(\cdot)$ assesses the candidate solution's quality. In Equation (13) and Equation (14), X_i^t and X_i^{t+1} represent the current and next positions of the i^{th} solution at iteration t, while X_{best}^t denotes the best solution obtained. The parameter r is the control coefficient that regulates the search behavior, while $rand()$ introduces randomness to enhance exploration and avoid local optima and the selection mechanism ensures that the updated solution X_i^{t+1} replaces X_i^t only if it yields a better fitness value, i.e., $f(X_i^{t+1}) < f(X_i^t)$; otherwise, the previous solution is retained.

Algorithm 1: BSS-FFNN-Based Cloud Resource Optimization

$Input: D \rightarrow Dataset, MaxIter \rightarrow Maximumiterations, N \rightarrow Populationsize, lb, u \rightarrow Lowerandupperbounds$
$Output: M \rightarrow Trained\ FFNN\ model\ and\ optimal\ configuration\ Xbest$
$Begin$
1. $Load\ dataset\ D$
2. $Preprocess\ data$:
   $Apply\ Min - Max\ normalization\ and\ IQR$
3. $Feature\ extraction\ using\ PCA$:
   $N^{m*m} = n_{j,i}, (n_{j,i} = Cov(B_j, B_i))$
   $Select\ top\ eigen\ vectors \rightarrow Feature\ set\ F$
4. $Initialize\ FFNN\ and\ BSS\ population$:

$O_i = f(X_i)$
5. $For\ t = 1\ to\ MaxIter\ do$:
$Update\ BSS\ position$:
$X_i^{t+1} = X_i^t + r \cdot (X_{best}^t - X_i^t)$
$Selection\ rule$:
$X_i^{t+1} = X_i^{t+1}, if\ f(X_i^{t+1}) < f(X_i^t)$
$else \quad X_i^t$
$Evaluate\ FFNN\ fitness\ and\ update\ Xbest$
6. $Return\ trained\ model\ M\ and\ optimal\ solution\ Xbest$
End

Algorithm 1 outlines a BSS-FFNN-based cloud resource optimization for better workload management. The process commences by performing preprocessing of data where the IQR method is used to eliminate outliers and Min-Max normalization is applied for feature scaling. Next, PCA is done for reducing the dimensions. The FFNN predicts the system's performance, while the BSS algorithm tunes resource allocation parameters through iterations. Finally, the selection of the best solution occurs after fitness evaluation.

4. Result and discussion

This section gives experimental results of the suggested BSS-FFNN model. To evaluate the BSS-FFNN algorithm's performance and compare it with benchmark models like CF-DNN [16], a whole experiment was set up using the Python programming language. Some of the parameters used for evaluation purposes include Response Time, Throughput, CPU Utilization, and Memory Utilization.

4.1 Cloud Resource and Performance Analysis

The relationships of the normalized cloud systems' features used to evaluate CPU utilization, memory utilization, network utilization, response time, and throughput, which have been normalized to 0 – 1 range, are depicted in Figure 2. Higher clustering is observed in case of network utilization near 0.8 -1.0, while most of the clustering points for response time and throughput lie below 0.5. The variations of these feature relationships with respect to the state of the system reflect the role played by resource utilization in terms of system performance.

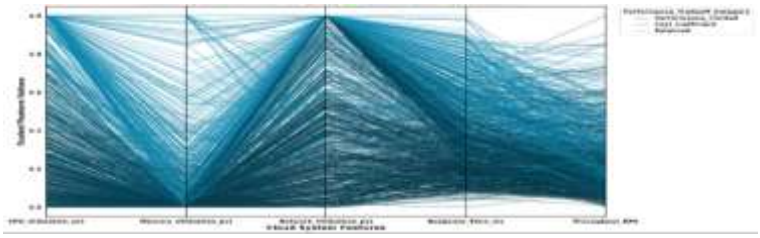

Figure 2: Multi-resource utilization analysis.

The cloud workload characteristics and resource are depicted in Fig. 3 where significant parameters include the number of active users, CPU utilization and resource utilization with respect to time. As seen in Figure 3(a), the active user counts from about 500 to 22,000 while response times span from 100 ms to 1,200 ms, exhibiting a high positive correlation, where as the active users increase, so do the response times mainly focused in the 2,000–12,000 active users with 300–800 ms response times. Figure 3(b) displays CPU utilizations of 10%–100% with respective response times also in the range of 100 ms to 1,200 ms where an increase in the CPU utilizations is associated with higher response delays mainly within 10%–30%. Figure 3(c) depicts trends in the average resource utilization with respect to time showing relatively constant allocations where CPU usage is around 25%–30%, memory usage at 40%–45%, network usage at 20%–25% and disk usage at 35%–40%.

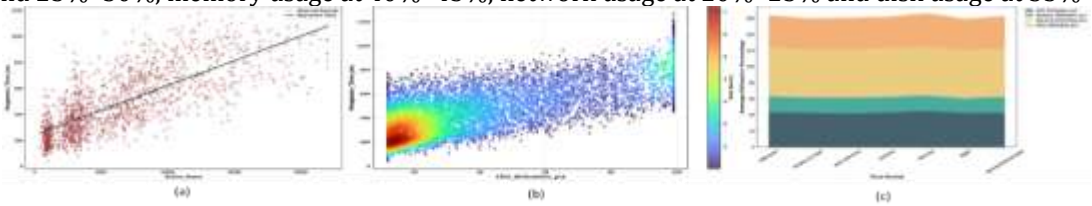

Figure 3: Analysis of (a) User load and response-time analysis. (b) CPU utilization impact analysis. (c) Resource utilization trend analysis.

4.2 Metrics explanation

Response Time (ms): This means the duration that the computer takes to react to the command of a user or task processing. The shorter the response time, the higher the efficiency of the system, and this is important in many fields. **Throughput (req/sec**): It describes the amount of requests that a computer can process within one second. A high throughput signifies that a system is efficient and can process many jobs at once. **CPU Utilization (%):** This is the percentage of CPU usage during task processing. It assists in measuring how efficient and effective a computer can perform tasks. Memory Utilization (%): This refers to the percentage of the RAM utilized in the processing of the workload in a computer.

4.3 Performance evaluation

The suggested model is based on data that has already been gathered from several sources, including transactions, logs, and demographics. As per Table 2, the performance of the BSS-FFNN is superior to that of the CF-DNN under all workloads [16]. For instance, in the presence of moderate to extremely high load, BSS-FFNN responds faster to requests with reduced response times and higher throughputs. Additionally, the system has more scalability. It uses fewer CPUs and memory, which implies that there is effective resource utilization. The new model performs very well even under heavy load conditions, thus showing the efficiency of using BSS optimization technique together with FFNN forecasting method.

Table 2: Analyzing the performance of the proposed model with the existing dataset

Load Level	Response Time (ms)	Throughput (req/sec)	CPU Utilization (%)	Memory Utilization (%)	Response Time (ms)	Throughput (req/sec)	CPU Utilization (%)	Memory Utilization (%)
	CF-DNN [16]				BSS-FFNN [Proposed]			
Normal Load	45	520	55	40	38	550	50	36
High Load	110	780	75	65	90	840	70	60
Peak Load	210	990	90	85	170	1080	84	78
Extreme Load	380	890	98	95	290	1030	90	87

Both the previous dataset and the suggested one are learned through the BSS-FFNN. Based on Table 3 and Figure 4, the BSS-FFNN model is more effective compared to the previous CF-DNN [16] model in all workload cases. Lower response time and higher throughput rates in the BSS-FFNN model indicate efficiency and scalability of the system. Additionally, lower CPU and memory usage rates also indicate efficiency in resource consumption. Such improvement can be observed in the Low Load and Flash Load workloads.

Table 3: Comparison of existing and proposed models trained on the cloud performance dataset

Load Level	Response Time (ms)	Throughput (req/sec)	CPU Utilization (%)	Memory Utilization (%)	Response Time (ms)	Throughput (req/sec)	CPU Utilization (%)	Memory Utilization (%)
	CF-DNN				BSS-FFNN [Proposed]			
Normal Load	42	525	53	38	36	600	49	35
High Load	105	790	73	64	88	900	68	58
Low Load	32	340	35	28	24	400	29	22
Flash Load	380	890	98	95	275	1080	89	84

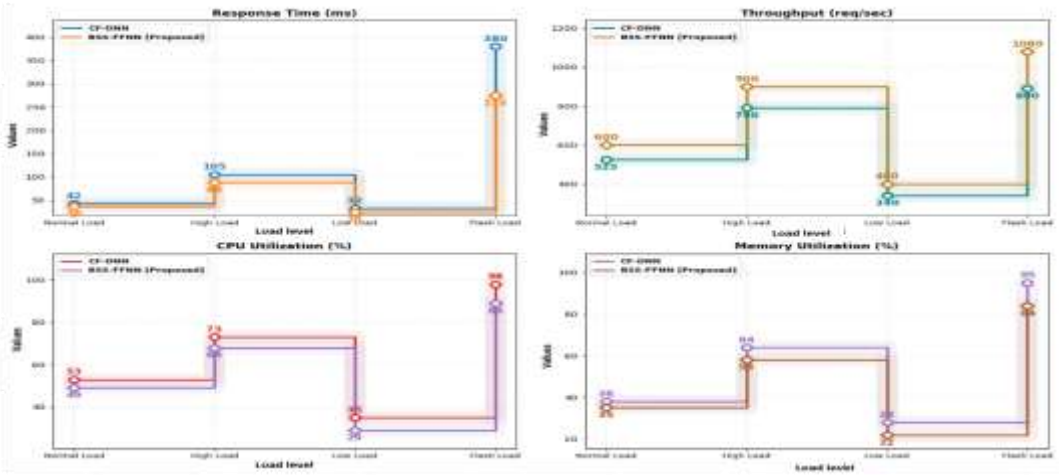

Figure 4: Representation of the existing and proposed models trained on the cloud performance dataset

Discussion: The existing model [16] achieves good scalability and accuracy using microservices, containerization, and hybrid ML techniques, but it suffers from high response time under extreme load, inefficient resource utilization, and limited adaptability to dynamic workloads. However, the proposed solution is based on the prediction process via FFNN along with BSS optimization and improved data preprocessing that makes it possible to perform more efficient and flexible resources allocation. Consequently, it allows reducing the response time, increasing resource efficiency, enhancing SLA compliance, and achieving better performance-cost ratio.

5. Conclusion

The suggested BSS-FFNN model has successfully managed the cloud-based e-commerce workload through prediction of system performances and optimization of resource allocation configuration. In doing so, it ensures the balance between system performance and system costs. To enhance the performance of the model, Min-Max normalization, outlier elimination based on IQR, and PCA have been applied. With its response time at 36 ms, throughput at 600 req/ms, CPU usage rate at 49%, and memory usage at 35%, it is implemented in Python language with Deep Learning library. Nevertheless, there are difficulties encountered in managing extremely dynamic and scalable cloud environment. Future works need to investigate adaptive multi-cloud optimization, reinforcement learning-based decision-making, and workload prediction & scheduling.

Reference

1. Daruvuri, R., Patibandla, K., and Mannem, P., 2024. Leveraging unsupervised learning for workload balancing and resource utilization in cloud architectures. International Research Journal of Modernization in Engineering Technology and Science, 6(10), pp.1776-1784. https://www.doi.org/10.56726/IRJMETS62304
2. Kandregula, N., 2022. Evaluating performance and scalability of multi-cloud environments: Key metrics and optimization strategies. World Journal of Advanced Research and Reviews, 15(01), pp.842-857. https://doi.org/10.30574/wjarr.2022.15.1.0560
3. Dritsas, E. and Trigka, M., 2025. Machine learning in e-commerce: Trends, applications, and future challenges. IEEE Access,13, pp.99048-99067. https://doi.org/10.1109/ACCESS.2025.3572865
4. Ramamoorthi, V., 2025. Advances in AI and ML for Cloud Computing: A Review of Algorithms, Challenges, and Innovations. International Journal of Scientific Research in Science and Technology, 12(5), pp.60-73. https://doi.org/10.32628/IJSRST2513120
5. Akinbolajo, O., 2023. Intelligent load balancing and concurrency control in cloud-based distributed databases: A machine learning approach 09(01), pp.847-854. https://doi.org/10.30574/ijsra.2023.9.1.0350

6. Machiraju, V.S., Kumar, V. and Sharma, S., 2026. ML-Based Autoscaling for Elastic Cloud Applications: Taxonomy, Frameworks, and Evaluation. Mathematical and Computational Applications, 31(2), p.49. https://doi.org/10.3390/mca31020049
7. Jin, L. and Chen, L., 2024. Exploring the impact of computer applications on cross-border e-commerce performance. IEEE Access, 12, pp.74861-74871. https://doi.org/10.1109/ACCESS.2024.3385017
8. Kostopoulos, G., Stefani, A., Vasiliadis, V. and Kotsiantis, S., 2026. Deep Learning for e-Commerce: Recent Developments in Prediction, Personalization, and Decision Intelligence. Applied Sciences, 16(5), p.2263. https://doi.org/10.3390/app16052263
9. Prasad, V.K., Dansana, D., Bhavsar, M.D., Acharya, B., Gerogiannis, V.C. and Kanavos, A., 2023. Efficient resource utilization in IoT and cloud computing. Information, 14(11), p.619. https://doi.org/10.3390/info14110619
10. Kalkha, H., Khiat, A., Bahnasse, A. and Ouajji, H., 2023. The rising trends of smart e-commerce logistics. IEEE Access, 11, pp.33839-33857. https://doi.org/10.1109/ACCESS.2023.3252566
11. Fé, I., Matos, R., Dantas, J., Melo, C., Nguyen, T.A., Min, D., Choi, E., Silva, F.A., and Maciel, P.R.M., 2022. Performance-cost trade-off in auto-scaling mechanisms for cloud computing. Sensors, 22(3), p.1221. https://doi.org/10.3390/s22031221
12. Licardo, J.T., Tanković, N., Osman, I., Lorencin, I., and Baressi Šegota, S., 2026. Performance Trade-Offs of Optimizing Small Language Models for E-Commerce. Big Data and Cognitive Computing, 10(5), p.155. https://doi.org/10.3390/bdcc10050155
13. Garg, V. and Jain, A., 2024. Scalable Data Integration Techniques for Multi-Retailer E-Commerce Platforms. International Journal of Computer Science and Engineering, 13(2), pp.525-570. http://www.iaset.us/
14. Omoike, N.O., 2023. Designing a secure and high-performing e-commerce platform for public cloud. International Journal of Science and Research Archive, 9 (2), 1008–1013. https://doi.org/10.30574/ijsra.2023.9.2.0525
15. Kodakandla, P., 2023. Real-Time Data Pipeline Modernization: A Comparative Study Of Latency, Scalability, And Cost Trade-Offs In Kafka-Spark-Bigquery Architectures. International Research Journal Of Modernization In Engineering Technology And Science, 5, pp.3340-3349. https://www.doi.org/10.56726/IRJMETS45355
16. Khurana, R., 2023. Hyper-Scalable Cloud-Native AI Frameworks for Predictive Hyper-Personalization in High-Traffic E-Commerce Ecosystems. INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY, 14, pp.196-206. https://iaeme.com/Home/issue/IJCET?Volume=14&Issue=3