International Journal of Intelligent Data and Machine Learning (IJIDML)
eISSN: 3087-4262
Volume03 Issue11 | Page No: - 01-11

Received: 22 Oct. 2026, Revised: 30 Oct 2026, Accepted: 03 November 2026, Published: 11 November 2026

DOI: https://doi.org/10.55640/ijidml-v03i11-01

Enhancing Cloud Resource Management Through Machine Learning-Based Utilization Prediction

Vivek Sharma
Designation: System Analyst, MITS Deemed University, India

ABSTRACT: Cloud resource management is important in the provision of efficiency in the allocation, scalability, and performance of virtualized computing environments. The ability to predict resource consumption is a very important and highly effective machine learning method that supports cloud operations in a perfect way by predicting future resource usage. An improved method for analyzing Microsoft Azure Trace data for forecasting is proposed in this research called BiLSTM-GRU, a deep learning architecture that combines Bidirectional Long Short-Term Memory with Gated Recurrent Units. The process for producing high-quality time-series inputs includes thoroughly cleaning the data, reducing noise, engineering features, and normalizing. BiLSTM captures temporal dependencies in both directions while GRU sequentially learns efficiently via gated operations, thus enabling both short- and long-term workload trend capture by the model. The LiR model outperforms the other linear resource trend models in terms of R^2 and RMSE (0.0169) according to the experimental data, while the BiLSTM-GRU model boasts the greatest prediction accuracy (0.9832%). The hybrid model's detection of the complex temporal variations in cloud workloads is further validated by a comparative analysis against the current NN, BP, and ARIMA models. The BiLSTM-GRU model is not only resourceful but also stable and offers correct CPU usage predictions, which enhance the efficiency of the cloud.

KEYWORDS: Cloud Resource Management, Machine Learning, Utilization Prediction, Cloud Computing Optimization, Resource Allocation, Predictive Analytics.

1. Introduction

The digital era's rapid development has completely changed the way companies and people relate to technology, with cloud computing being one of the most disruptive innovations among these [1][2]. It enables users to pay every month for the computing resources they require under its various access offerings such as software, platforms, and infrastructure, thereby providing the most up-scaled, elastic, and cheapest solutions [3][4]. This transition has influenced every sector of the business world by making the business more efficient, helping to expand the digital world as far as possible, and even leading to the emergence of new concepts in the respective spheres [5][6]. Simultaneously, the cloud ecosystems are highly dynamic, and their inability to be predictable comes with a steep price of being unable to effectively control the resources, which is an inadmissible requirement of maintaining the performance and cost efficiency at the necessary levels.

Cloud computing provides a flexible platform in which applications can receive the resources they desire immediately before executing them on Virtual Machines (VMs) [7][8]. Most service providers choose the pay-as-you-go approach since it gives customers greater leeway and cheaper rates [9][10]. There is a growth in the quantity of cloud users and applications to an extent where there has been a demand to have efficient utilization and management of resources [11][12]. Since cloud workloads alter a loT, highly proficient resource management is a necessity to address the evolving user demands [13]to maximize resource utilization and to maintain a stable system [14][15]. The main aspects of attaining these goals are the prediction procedures for utilizing resources.

The dynamic nature of cloud computing is that it dynamically manages and distributes the computational power [16][17]. Scalability and flexibility of the cloud enabled real-time suppliers to allocate resources at the same speed as the changing workloads, thereby boosting operability and agility [18][19][20]. Nevertheless, conversely, this dynamic feature has added complexity to the allocation of resources [21], and that is why they require advanced placement strategies, which can easily adapt to the evolving nature of the cloud [22][23][24].

The drawbacks of cloud computing have resulted in the creation of a novel and efficient resource distribution model that includes ML and DL approaches as methods to optimize cloud performance and scalability [25][26][27]. Precise forecasting, real-time updates, and effective resource allocation are made possible by the solution's usage of deep learning for demand prediction and reinforcement learning for dynamic scheduling [28][29][30]. To convince them of the quality and longevity of cloud computing environments, this framework is designed to maximize resource utilization, decrease latency and costs, and improve overall efficiency and sustainability [31].

A. Motivation and Contribution

Two key factors that led to the creation of this specific research are the increasingly complex nature of cloud environments and the fact that highly capable, intelligent, and automated resource management systems are needed. Traditional ML models struggle to keep up with the ever-changing and time-sensitive cloud workloads, leading to ineffective resource usage and performance degradation. Therefore, a complex predictive model that has the capability of detecting intricate temporal trends and making real-time resource-allocation decisions with high precision is desperately needed to solve these problems. In order to improve the cloud system's overall efficiency, reduce latency, and increase prediction accuracy, the suggested Hybrid RNN+GRU model captures the data's short-term and long-term relationships extremely well. This research offers several key contributions as listed below:

- This article proposes an end-to-end method that uses a shared framework to integrate data preparation, feature engineering, model construction, and performance evaluation utilizing a specific dataset in Microsoft Azure.
- Bidirectional LSTM and GRU layers are integrated to produce a new hybrid deep learning framework that enables the model to adequately comprehend both dynamic changes in workload and forward-backward temporal dependencies.
- A comparison with baseline models like Linear Regression (LiR), ARIMA, BP, and NN is used to assess the predictive performance of the hybrid model. When it discusses R2 accuracy and dependability in time-series forecasting, the hybrid model performs better than the others.
- Comprehensive performance evaluation is carried out using RMSE, MSE, MAE, and R^2 score, which enables standardization and interpretation of the comparison of the prediction models.
- The recently proposed BiLSTM-GRU hybrid model provides extremely accurate CPU utilization estimates, leading to more effective cloud resource allocation, less computational overhead, and improved workload scheduling techniques.

B. Justification and Novelty

The ability to handle the extremely dynamic and nonlinear patterns of cloud resource usage, where traditional forecasting models might not be able to capture the complex time dependencies, leading to unacceptably high inaccuracies, is the basis for this research's justification and novelty. Long-term trends and short-term variations cannot be predicted by either pure deep learning or classic machine learning methods, which leads to erroneous forecasts and ineffective resource allocation. The proposed Hybrid BiLSTM+GRU model combines the gating lightness of GRU with the two-way learning capability of BiLSTM, enabling deeper comprehension of the temporal data and faster convergence. In addition to enhancing the model's scalability, adaptability, and robustness in actual cloud systems, this hybrid architecture also increases the model's ability to learn intricate workload behavior and offers extremely accurate CPU utilization forecasts along with an overall high-performance level.

C. Organization of the Paper

An overview of the literature and related studies in Section II. In Section III, the description of the dataset, methods of preprocessing the data, and the evaluation of features are described. Section IV is the implementation, and Section V is the experimental results, and a performance comparison with the existing models is presented. Section VI provides the conclusion and future work.

2. Literature Review

Recent studies on Cloud Resource Management were performed through a detailed review and analysis to guide and support the work. Table I illustrates the major research in this field, its proposed models, data applied, major findings, and the limitations found in each method.

A. Javeed, et al. (2026) Customer service requests (TPS) are predicted using a two-stage forecasting model, which then uses cascaded XGBoost models to estimate CPU burden. Concept drift is addressed via adaptive online retraining using an expanding-window approach. The model obtained SMAPE below 7% for the majority of apps when tested on private cloud traces from 10 applications; the top application recorded MAE of 0.7372, RMSE of 1.1866, SMAPE of 3.57%, and R^2 of 0.9185. Stable recursive forecasting with controlled error buildup over 60 prediction steps was validated by horizon-wise analysis [32].

Han (2025) supports both fixed and dynamic

assignments of tasks, and thus is flexible to different applications. The FSO algorithm would also ensure that an adequate balance between exploitation and exploration is maintained and hence prevent early convergence. Overall, it fosters greater performance and cost-efficiency of cloud operations. The scalability and flexibility of this method prove its capabilities of this method in modern cloud infrastructure with accuracies, precisions, recalls, and F1-Scores of 98 % [33].

Doukha and Ez-Zahout (2025) installed PROXMOX and Prometheus to monitor and Grafana to visualize and give alerts. Through the usage of ML models, such as RF Regression and LSTM, the framework models the usage of resources based on history, which allows accurate and timely allocation of resources. It shows that the Random Forest model has a better accuracy (MAPE of 2.65) and is far better than LSTM (MAPE of 17.43). Such results support the applicability of Random Forest in estimating resources [34].

Wang and Yang (2025) suggest a smart resource allocation system that utilizes deep learning (LSTM) in the prediction of demand and reinforcement learning (DQN) in the dynamic scheduling. The proposed system improves resource utilization (32.5), minimum response time (43.3), and cost of operation (26.6) by accurately predicting the needs of computing resources and providing them in real time. The approach greatly increases efficiency while preserving high service quality, according to experimental results in a production cloud environment [35].

Dixit et al. (2024) assessed DNNs, RF, and gradient-boosting machines on different datasets and complex tasks. Focusing on the obtained results, it becomes possible to state that the proposed methods yield up to 87% improvement in the training speed and a 5–12% increase in the accuracy of the models compared to the traditional implementations on single nodes. Moreover, the approach proved its capability of scalability, did not degrade the performance, and achieved near-linear speedup throughout the distributed nodes of up to 1000 [36].

Ramya, Ramya, and Babu Rao (2023) The suggested method achieved an accuracy of 19.49%, 18.05%, 12.77%, and 7.89%, respectively, outperforming a number of previous methods, including DL, TensorFlow DL, Cuckoo search-based NN, and ML. In addition to better performance metrics, the new technique achieved minimal energy of 0.301 J, minimal latency of 0.666 sec, maximal accuracy of 0.950, maximal sensitivity of 0.934, and maximal specificity of 0.967. Therefore, the created FrHHGO-based ShCNN e-waste classification method improves emerging economies' social, environmental, and economic sustainability [37].

Table 1. The Summery of the Recent Studies on Cloud Resource Management using Machine Learning

Author	Proposed Work	Results	Key Findings	Limitations & Future Work
A. Javeed et al. (2026)	Two-stage integrated forecasting using cascaded XGBoost models to predict TPS and CPU workload, with adaptive online retraining using an expanding-window strategy.	SMAPE below 7% for most applications. Best application: MAE 0.7372, RMSE 1.1866, SMAPE 3.57%, and R^2 0.9185.	Achieved accurate workload forecasting and stable recursive predictions with controlled error accumulation across 60 prediction steps.	Evaluated on private cloud traces from ten applications. Further validation on diverse cloud environments and larger workloads is required.
Han (2025)	Introduced a Fish Swarm Optimization (FSO) algorithm enabling static and dynamic task assignment for adaptive cloud operations.	Achieved 98% accuracy, precision, recall, and F1-score.	Ensures balance between exploration and exploitation, improving scalability and cost-effectiveness.	Future work could explore hybridization with deep learning models for further optimization.
Doukha and Ez-Zahout (2025)	Developed a predictive framework using Random Forest Regression and LSTM integrated with PROXMOX, Prometheus, and Grafana for real-time monitoring.	Random Forest achieved MAPE of 2.65%, outperforming LSTM's 17.43%.	Demonstrated the superiority of Random Forest in accurate resource usage prediction.	Future research could include real-time workload adaptation and multi-cloud scalability.
Wang and Yang (2025)	Proposed a hybrid deep learning approach combining LSTM for	Improved resource utilization by 32.5%, reduced	Significantly enhanced efficiency and service quality in	Can be expanded to include multi-agent RL systems for

	demand prediction and DQN for dynamic resource scheduling.	response time by 43.3%, and cut costs by 26.6%.	production environments.	distributed cloud frameworks.
Dixit et al. (2024)	Evaluated DNN, Random Forest, and Gradient Boosting Machines for distributed cloud training and scalability.	Achieved up to 87% improvement in training speed and a 5–12% accuracy increase.	Proved scalability with near-linear speedup across 1000 nodes.	Future work may focus on optimizing communication overhead in distributed learning.
Ramya, Ramya, and Babu Rao (2023)	Proposed a FrHHGO-based ShCNN model for e-waste classification and sustainability optimization.	Outperformed existing models with 95% accuracy, 0.301 J energy use, and 0.666 sec delay.	Improved environmental and economic sustainability using optimized neural architecture.	Future work could extend this model to cloud-based industrial waste management.

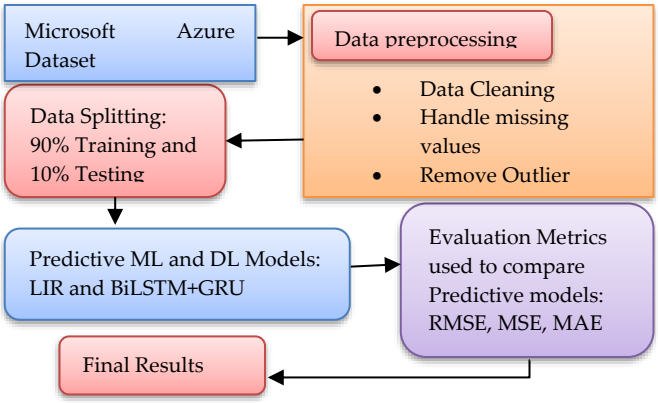


Fig. 1. Proposed Flowchart for CPU Utilisation for Microsoft Azure

3. Research Methodology

Cloud resource utilization prediction is an important task for efficient resource management and performance optimization. The proposed approach uses the Microsoft Azure dataset, a comprehensive log of cloud resource utilization, to predict CPU usage accurately. The data undergoes preprocessing, including data cleansing, handling missing values, removing outliers, and feature engineering to generate informative attributes. Linear Regression (LiR) and the BiLSTM-GRU hybrid model use the processed data to capture linear trends and complex temporal relationships. The dataset is divided into training and testing subsets in a 90:10 ratio to ensure proper model training and validation. After training, model performance is evaluated using RMSE, MSE, MAE, and R^2 to assess prediction accuracy and reliability. This evaluation helps identify the most suitable model for predicting CPU utilization during dynamic cloud activities. The overall workflow of the proposed method is presented in Figure. 1.

Each stage of the recommended approach is thoroughly explained in the following section. Each phase is explained in detail to highlight its significance and support the model creation process as a whole:

A. Data Gathering and Analysis

The Microsoft Azure Traces dataset provides CPU usage data that can be utilized for predicting CPU utilization and forecasting network transmission throughput. Important resource measures, such as CPU consumption, are available in the datasets, which are vital for the research's prediction models. To keep everything consistent, make comparisons easier, and reduce the range of the resource utilization measures in these datasets to 0 to 1. With each dataset's resource going through its own scaling-down procedure, the maximum capacity of each virtual machine may be represented by a value of 1. This standardization method for normalization distinguishes a uniform manner of dealing with resources' analysis and prediction across datasets that yield insights into the utilization patterns of resources that are strong and comparable in their robustness.

Figure 2. shows the CPU utilization trends in the Microsoft Azure environment over time. The plot illustrates variations in CPU usage and shows the highest (orange), lowest (blue), and average (green) CPU usage over the course of the report. The average line provides an approximate representation of the system performance over the monitoring period, indicating

stable and unstable times, while the many peaks and valleys indicate the fluctuating quantity of work completed.

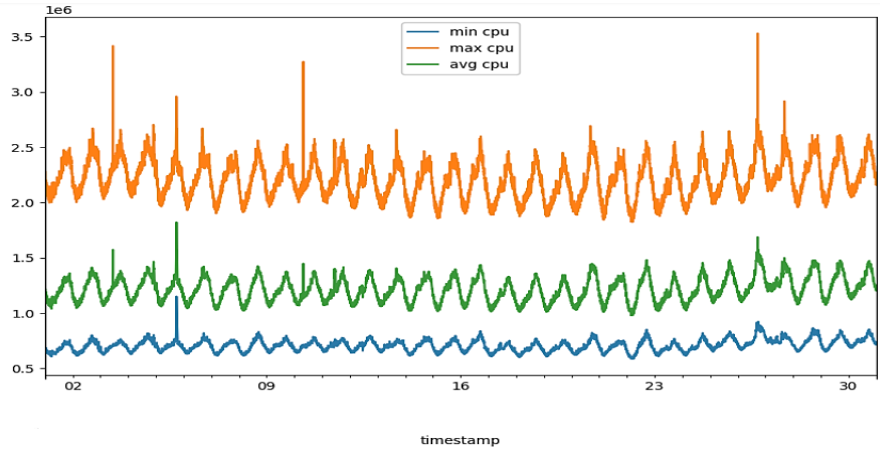


Fig. 2. Microsoft Azure Dataset CPU Utilisation

B. Data Pre-Processing

The dataset included a thorough data preparation procedure that comprised feature engineering, concatenation, and purification, among other steps. Missing value imputation, noise/outlier reduction, data labelling, and feature normalization were the primary preprocessing tasks. By supplying the necessary quality, consistency, and applicability of data for model training, all of the abovementioned preprocessing techniques greatly improved cloud resource management. The most important preprocessing steps have been briefly outlined below:

- **Data cleaning:** The trace data contains outliers and missing values, which are typically eliminated by removing those tuples from the dataset. Additionally, it is a truth that hardly every feature is useful for forecasting.
- **Handle missing values:** There are essentially two approaches to handling missing values. One option is to remove the incomplete records, while another is to fill them using statistical techniques like mean, median, or mode imputation. Along with ensuring data consistency, approaches to dealing with missing data can also help in avoiding errors in model training.
- **Remove Outliers:** The step is aimed at removing irrelevant or wrong data that may alter the model learning. Smoothing, filtering, and outlier detection are included among techniques to ensure high-quality data and improvement of model accuracy.

C. Feature Engineering

The process of turning raw data from cloud resources into useful features that help the model learn better. In the process, the original hourly time-series data is transformed into daily representations, during which multiple significant statistical characteristics, including daily averages and maximum values, are computed. Those features simplify the process of determining the patterns of resource consumption and task intensity; each of the servers has a daily time-series that demonstrates how the resource consumption and application KPIs depend on each other. To enable more precise and reliable predictions regarding cloud resource utilization, the last step is to aggregate these processed time-series data records into a structured dataset per server.

D. Data Splitting

The dataset is divided into two parts, with 10% designated for testing and 90% for training. Afterward, the training data is used to build and optimize the model, enabling it to identify hidden relationships and patterns in the data.

E. Proposed Linear Regression (LiR)

The link between the input and dependent variables may be shown by linear regression, which finds a straight line with the minimum prediction error [38]. By using the following formula, represented as Equation (1), to calculate the projected CPU or network consumption, the model forecasts the use of cloud resources:

$$\hat{y}_t = \beta_0 + \beta_{1x1,t} + \cdots + \beta_n x_{n,t} \qquad (1)$$

In the given expression, {\hat{y}}_t denotes the resource utilization forecast at instantiation t, $\beta 0$ represents the constant term, $\beta 1,\beta 2,\ldots,\beta n$ depict the coefficients of the model, and $x1,t,x2,t,\ldots,xn,t$ signify the input features affecting the resource consumption. The model is based on linear relationships among the factors and the real utilization values.

F. Gated Recurrent Unit (GRU) Model

An LSTM network that can provide nearly the same predictive performance but is simpler and quicker to train is the GRU, or Gated Recurrent Unit. While LSTM employs distinct input and forget gates, GRU merges them into a single update gate Z_t and adds a reset gate R_t to regulate memory flow, allowing the network to select what to remember and what to forget. While the reset gate influences the unlearning of the previous state, the update gate controls the quantity of historical data that is preserved. This is a representation of Equation (2) concerning the update gate for capturing information:

$$Z_t = \sigma(W_Z * [h_{t-1}, X_t]) \quad (2)$$

Equation (3) for getting information from the reset gate is as follows.:

$$R_t = \sigma(W_r * [h_{t-1}, X_t]) \quad (3)$$

In this case, {\bar{h}}_tis the output state of the unit at time t, and {\bar{h}}_tis the implied state value of h_t. Equation (4) describes the computation of candidate activation, which is provided by:

$$\bar{h}_t = tanh(W_{\bar{h}} * [r_t * h_{t-1}, X_t]) \quad (4)$$

and the final hidden state is updated using Equation (5):

$$h_t = (1 - Z_t) * h_{t-1} + Z_t * \bar{h}_t \quad (5)$$

The previous hidden layer, {\bar{h}}_{t-1}is employed in the preceding equation since the state X_t represents the current input. tanh is the hyperbolic tangent function, \sigma is the sigmoid-type activation function, and W_Z, W_r W z, and W_{\bar{h}} are matrix weights.

G. Proposed Hybrid Bilstm and Gated Recurrent Unit (Bilstm+GRU) Model

The Hybrid BiLSTM+GRU model is an amazing fusion of the BiLSTM network, which can learn from both directions in a sequence, and the efficient gating mechanism of the GRU [39]. The product is a model that is very successful at forecasting cloud resource utilization because it not only detects temporal relationships in both forward and backward directions but also retains important information through gated updates.

Together, the input sequence is scanned both forward and backward. Equation (6) is used to generate two hidden states at each time step:

$$\overrightarrow{h_t}, \overleftarrow{h_t} = BiLSTM(x_t) \quad (6)$$

In this instance, \vec{h_t} and htover left arrow \ t\ correspond to the hidden states moving forward and backward, respectively, while x_t represents the current input during time t. The model can simultaneously extract short-term patterns from past contexts and future ones thanks to this type of bidirectional organization.

The GRU layer receives the output from the concatenated BiLSTM and uses its gating mechanism, which is defined by Equation (7), to improve and modify the features:

$$h_t^{GRU} = (1 - Z_t) * h_{t-1}^{GRU} + Z_t * \bar{h}_t \quad (7)$$

In this equation, {\bar{h}}_t is the hidden state produced by the GRU that is a potential output, andZ_t functions as an update gate that controls the ratio of old memory to fresh data. The hybrid model can capture long-term interdependence at a lower computational cost thanks to the gating mechanism.

H. Evaluation Metrics

Since the R^2 score, RMSE, MAE, and MSE are the most appropriate measures for the accuracy of predictions in continuous time-series datasets, they were used in the study to assess the predictive performance of the new model aimed at predicting CPU utilization. These standard performance metrics were used to assess the efficacy of ML and DL models with baseline and hybrid algorithms after they had been trained and tested. Equations (8)–(11) illustrate the manner in which these fundamental characteristics provide the basis for measuring each model's overall prediction power.

1) Root Mean Squared Error (RMSE)

RMSE is calculated by determining the squared differences between the actual values and the predicted ones, taking the mean of those squared differences, and in the end, taking the square root of the mean. RMSE is capable of predicting errors with higher accuracy than MAE. Therefore, it becomes a handy instrument for identifying outliers or major differences:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y)^2} \quad (8)$$

Where n= “data points”, $\hat{y}_i$= “predicted values”, y_i= “actual values”.

2) R-squared (R2) Score

The R2 value shows how well the independent factors can forecast changes in the dependent variable. The essence of which is derived from taking the explained variance in the numerator and the total variance in the denominator in order to obtain a ratio. The formula for the R2 score is:

$$R^2 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (9)$$

Here, n is the number of samples, y_i is the actual value, and {y_i}^2 is the predicted value.

3) Mean Squared Error (MSE)

The average of the squares of mistakes or deviations is how MSE is computed. The computation process involves averaging the squared discrepancies between the actual and anticipated values. The MSE formula is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (10)$$

Here, y_i represents the actual value, ${y_i}^2$represents the predicted value, and n is the number of samples.

4) Mean Absolute Error (MAE)

The MAE is calculated by averaging the absolute values of the forecast and the actual results. It takes into account the average size of mistakes but does not take their direction into account. One way to calculate MAE is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (11)$$

In this case, y_i is the actual value, {y_i}^2 is the predicted value, and n is the number of samples.

4. Results and Discussion

The section below describes the experimental design and gives an overview of how the proposed model performs during training and testing to reveal its computational efficiency and effectiveness. The hardware of the experimental platform is now made up of an Intel Core i7-6500U processor, 8 GB and 1 TB of memory and storage. Table II analyzes the cloud resource performance of the proposed BiLSTM-GRU hybrid model and Linear Regression (LiR) model. Besides the lowest RMSE of (0.0169), MAE (0.0131), and MSE (0.0002), the LiR model had an R^2 of (0.9733), which means that it was very well predicted linearly. On the other hand, the BiLSTM–GRU model achieved the maximum accuracy with an R2 of 0.9832. The model, despite having RMSE (0.0214) and MAE (0.0169) that are very marginally higher than those of LiR, is able to detect complex temporal dependencies, which are typically found in cloud workload forecasting within changing environments. Predictive stability of the model remains excellent as reflected in an MSE of 0.0004, even though time-series patterns are highly variable.

Table II. Classification Results of the Proposed Hybrid Model for Cloud Resource Management

Matrix	LiR	BiLSTM+GRU
R^2 score	0.9733	0.9832
RMSE	0.0169	0.0214
MAE	0.0131	0.0169
MSE	0.0002	0.0004

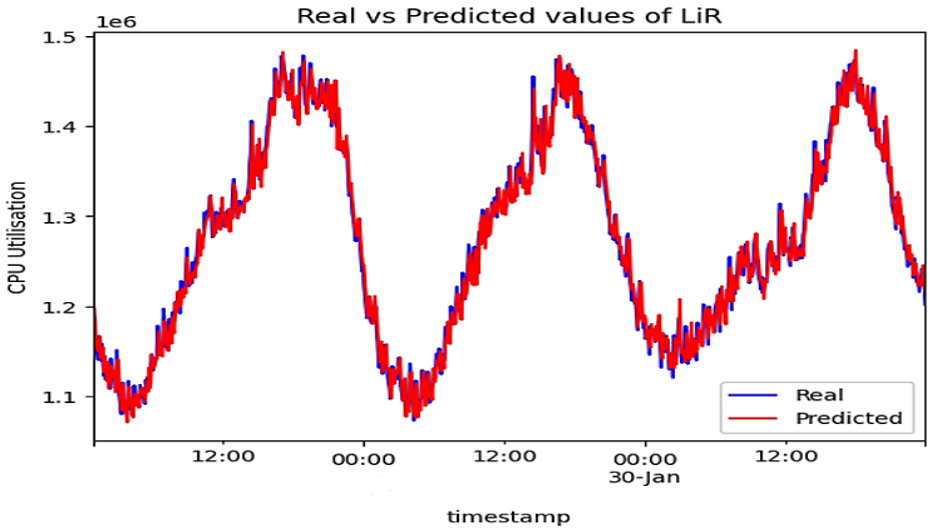


Fig. 3. Prediction on the Lir Model of Real vs Predicted Values

Figura 3. CPU Utilization Reale vs Predetto mediante Regressione Lineare (LiR). The graphic gives a simultaneous perspective of the real and forecasted CPU utilization figures over the study period. The Linear Regression model manages to follow the CPU usage trend, depicting its increases and decreases accurately. The model does not completely lose track of the general time pattern, although there are small discrepancies caused by the fast fluctuation periods. The fact that the model can replicate the overall time trend of resource consumption is indicative that the model can be utilized as a baseline estimator of cloud resource management.

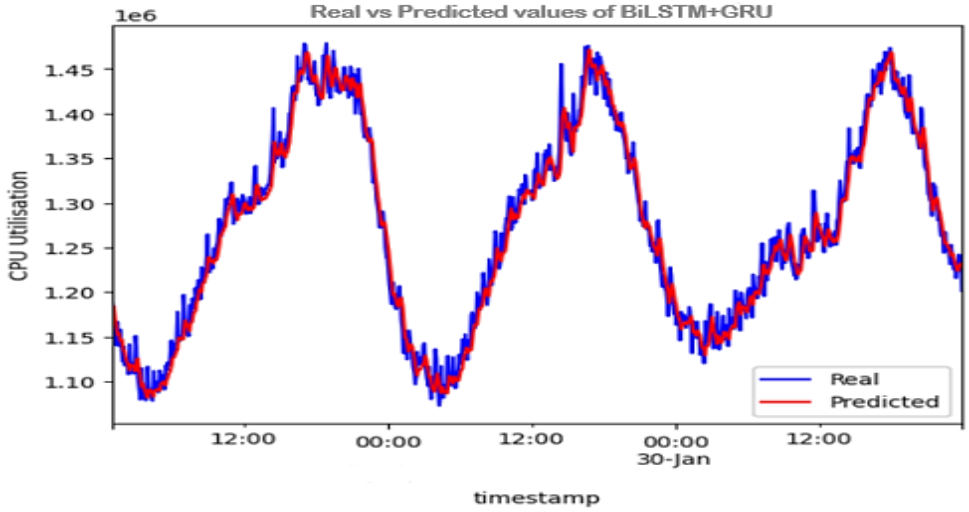


Fig 4. Prediction on BiLSTM–GRU Hybrid Model for Actual vs. Predicted Values.

Figure 4. depicts the variation in the actual and predicted CPU utilization with the BiLSTM-GRU hybrid model. The graphic explains that throughout the duration of the period, the real and predicted numbers of the CPU utilization values are very close to each other. The deviation of the real numbers is not that significant since the BiLSTM-GRU model is able to distinguish the periodic patterns and sudden fluctuations of the workload. The model's exceptional capacity to learn intricate temporal connections is demonstrated by how well it tracks peaks and valleys, making it an ideal choice for precise forecasting of cloud resource utilization.

A. Comparative Analysis

To evaluate whether the model is effective, a comparative analysis using R2 and RMSE with other models is shown in Table III. A comparison of several ML and DL models designed to predict CPU usage is shown in Table III. With an R^2 value of 0.952322 and an RMSE of 0.025668, the Neural Network (NN) model performs the best, whereas the Backpropagation (BP) and ARIMA models have substantially less predictive ability, with R^2 values of 0.8938 and 0.893, respectively. However, with an R2 of 0.9733 and an RMSE of 0.0169, the Linear Regression (LiR) model is unquestionably the best option. These numbers suggest that it is highly capable of performing the task of stretching the linear resource usage trend. On the other hand, the BiLSTM–GRU hybrid model has the maximum accuracy, as evidenced by an R2 score of 0.9832; hence, the model is said to capture complex temporal dependencies. Although the model's RMSE (0.0214) is marginally higher than LiR's, the hybrid model's temporal learning is so much better that it continues to be the best choice for CP utilization forecasts.

Table III. Comparison of Different Machine learning and deep learning Models for CPU Utilisation Prediction

Model	R^2 score	RMSE
NN[40]	0.952322	0.025668
BP[41]	0.8938	0.681
ARIMA[42]	0.893	0.157
LiR	0.9733	0.0169
BiLSTM+GRU	0.9832	0.0214

Several approaches for predicting CPU usage using DL and ML have been thoroughly examined. Despite having a decent forecasting capability, the traditional methods NN, BP, and ARIMA were unable to fully capture the complex temporal patterns of cloud workloads. However, given the greater linear trend over time, Linear Regression is the most effective model thus far. However, the proposed BiLSTM–GRU hybrid model has proven to be the best in every way because it not only learns complex sequential patterns but also makes reliable, highly accurate predictions. As a result, this model can be considered the most potent and appropriate for predicting CPU utilization in highly variable cloud environments.

5. Conclusion and Future Study

Cloud resource management has progressively become an important aspect of modern computing environments that strongly relies on high speed, scalability, and reliability. One of the tasks that acquires the greatest importance is the forecast of resource utilization, as it is the main key to the successful distribution and the prevention of a decline in performance. To achieve this, a machine learning-based system, which comprised all preprocessing, feature engineering, and model evaluation procedures, was

developed to predict CPU utilization on the Microsoft Azure dataset. Prediction accuracy was higher with the suggested Hybrid BiLSTM-GRU model than with either the conventional or deep learning models. In addition to the strong linear prediction power of Linear Regression, the hybrid design achieved the best R2 score of 0.9832 by revealing the intricate temporal relationships. The findings of this study demonstrate the use of hybrid sequence-learning models in cloud workload prediction, which can be very beneficial for intelligent cloud resource optimization.

Future studies would expand the technique to multi-resource prediction, enabling it to forecast CPU, memory, disc, and network utilization simultaneously, improving cloud performance management effectiveness. Virtual development and real-time data ingestion would likely significantly improve the system's capacity to handle abrupt changes in workloads.

References

1. Pezzuolo, V. Milani, D. Zhu, H. Guo, S. Guercini, and F. Marinello, "On-Barn Pig Weight Estimation Based on Body Measurements by Structure-from-Motion (SfM)," Sensors, vol. 18, no. 11, p. 3603, Oct. 2018, doi: 10.3390/s18113603.
2. Parupalli and H. Kali, "An In-Depth Review of Cost Optimization Tactics in Multi-Cloud Frameworks," Int. J. Adv. Res. Sci. Commun. Technol., vol. 3, no. 5, pp. 1043–1052, Jun. 2023, doi: 10.48175/IJARSCT-11937Q.
3. L. Wang and Y. Liao, "A short-term hybrid wind speed prediction model based on decomposition and improved optimization algorithm," Front. Energy Res., vol. 11, pp. 1–17, Nov. 2023, doi: 10.3389/fenrg.2023.1298088.
4. M. R. R. Deva and N. Jain, "Utilizing Azure Automated Machine Learning and XGBoost for Predicting Cloud Resource Utilization in Enterprise Environments," in 2025 International Conference on Networks and Cryptology (NETCRYPT), IEEE, May 2025, pp. 535–540. doi: 10.1109/NETCRYPT65877.2025.11102235.
5. Zhou and D. Fu, "Soft Measurement of Wind Tunnel Dynamic Flow Based on Attention_LSTM-Kalman Modeling," J. Data Acquis. Process., vol. 37, no. 2, pp. 463–470, 2022, doi: 10.16337/j.1004-9037.2022.02.019.
6. S. BuchiReddy Karri, C. M. Penugonda, S. Karanam, M. Tajammul, S. Rayankula, and P. Vankadara, "Enhancing Cloud-Native Applications: A Comparative Study of Java-To-Go Micro Services Migration," Int. Trans. Electr. Eng. Comput. Sci., vol. 4, no. 1, pp. 1–12, Apr. 2025, doi: 10.62760/iteecs.4.1.2025.127.
7. G. Kaur, A. Bala, and I. Chana, "An intelligent regressive ensemble approach for predicting resource usage in cloud computing," J. Parallel Distrib. Comput., vol. 123, pp. 1–12, Jan. 2019, doi: 10.1016/j.jpdc.2018.08.008.
8. R. Patel, "Advancements in Renewable Energy Utilization for Sustainable Cloud Data Centers: A Survey of Emerging Approaches," Int. J. Curr. Eng. Technol., vol. 13, no. 05, pp. 447–454, Oct. 2023, doi: 10.14741/ijcet/v.13.5.7.
9. Muteeh, M. Sardaraz, and M. Tahir, "MrLBA: multi-resource load balancing algorithm for cloud computing using ant colony optimization," Cluster Comput., vol. 24, no. 4, pp. 3135–3145, Dec. 2021, doi: 10.1007/s10586-021-03322-3.
10. Rahmanian, M. Ghobaei-Arani, and S. Tofighy, "A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment," Futur. Gener. Comput. Syst., vol. 79, pp. 54–71, Feb. 2018, doi: 10.1016/j.future.2017.09.049.
11. N. Malik, M. Sardaraz, M. Tahir, B. Shah, G. Ali, and F. Moreira, "Energy-Efficient Load Balancing Algorithm for Workflow Scheduling in Cloud Data Centers Using Queuing and Thresholds," Appl. Sci., vol. 11, no. 13, p. 5849, Jun. 2021, doi: 10.3390/app11135849.
12. Mason, M. Duggan, E. Barrett, J. Duggan, and E. Howley, "Predicting host CPU utilization in the cloud using evolutionary neural networks," Futur. Gener. Comput. Syst., vol. 86, pp. 162–173, Sep. 2018, doi: 10.1016/j.future.2018.03.040.
13. R. Duggasani, "Scalable and Optimized Load Balancing in Cloud Systems: Intelligent Nature-Inspired Evolutionary Approach," Int. J. Innov. Sci. Res. Technol., vol. 10, no. 5, pp. 2153–2160, May 2025, doi: 10.38124/ijisrt/25may1290.
14. Sugawara and H. Nikaido, "Auditoría informática, para la evaluación de riesgos en la seguridad de la información en la Cooperativa de Ahorro y Crédito PRODVISIÓN, de la Provincia de Tungurahua, Canton Pelileo," Trab. titulación, 2021.
15. P. Chandrashekar, "A Survey of Tools, Techniques, and Best Practices: CI/CD Integration in DevOps Workflows," Int. J. Adv. Res. Sci. Commun. Technol., vol. 3, no. 3, pp. 1366–1376, Jul. 2023, doi: 10.48175/IJARSCT-11978V.
16. T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, "Machine learning (ML)-centric resource

management in cloud computing: A review and future directions," J. Netw. Comput. Appl., vol. 204, p. 103405, Aug. 2022, doi: 10.1016/j.jnca.2022.103405.

17. S. R. Kurakula, "Cloud-native microservices in financial services: Architecting for scalability and flexibility," World J. Adv. Res. Rev., vol. 26, no. 2, pp. 1435–1442, May 2025, doi: 10.30574/wjarr.2025.26.2.1690.

18. Kumaran and E. Sasikala, "Computational access point selection based on resource allocation optimization to reduce the edge computing latency," Meas. Sensors, vol. 24, pp. 1–8, Dec. 2022, doi: 10.1016/j.measen.2022.100444.

19. P. Pradhan, P. K. Behera, and B. N. B. Ray, "Modified Round Robin Algorithm for Resource Allocation in Cloud Computing," Procedia Comput. Sci., vol. 85, pp. 878–890, 2016, doi: 10.1016/j.procs.2016.05.278.

20. Konda, "Performance Evaluation of Machine Learning Inference Workloads in Containerized Cloud Computing Environments," Eastasouth J. Inf. Syst. Comput. Sci., vol. 3, no. 02, pp. 292–311, Dec. 2025, doi: 10.58812/esiscs.v3i02.949.

21. Y. Macha and S. K. Pulichikkunnu, "A Survey of DevOps Practices for Machine Learning and Artificial Intelligence Workflows in Modern Software Development," ESP J. Eng. Technol. Adv., vol. 4, no. 3, pp. 200–208, 2024, doi: 10.56472/25832646/JETA-V4I3P121.

22. S. Mayur and N. Chaudhary, "Enhanced Weighted Round Robin Load Balancing Algorithm in Cloud Computing," Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 9S2, pp. 148–151, Aug. 2019, doi: 10.35940/ijitee.I1030.0789S219.

23. Alelyani, A. Datta, and G. M. Hassan, "Optimizing Cloud Performance: A Microservice Scheduling Strategy for Enhanced Fault-Tolerance, Reduced Network Traffic, and Lower Latency," IEEE Access, vol. 12, pp. 35135–35153, 2024, doi: 10.1109/ACCESS.2024.3373316.

24. Patil, V. Kansal, B. R, T. Manohar, D. Vekariya, and S. Kannan, "Energy-Efficient Resource Allocation for Smart Grids using Deep Reinforcement Learning in IoT Cloud Systems," in 2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, Oct. 2024, pp. 67–72. doi: 10.1109/I-SMAC61858.2024.10714834.

25. Basu, S. Kal, U. Ghosh, and R. Datta, "DRIVE: Dynamic Resource Introspection and VNF Embedding for 5G Using Machine Learning," IEEE Internet Things J., vol. 10, no. 21, pp. 18971–18979, Nov. 2023, doi: 10.1109/JIOT.2023.3235382.

26. Mitropoulou, P. Kokkinos, P. Soumplis, and E. Varvarigos, "Anomaly Detection in Cloud Computing using Knowledge Graph Embedding and Machine Learning Mechanisms," J. Grid Comput., vol. 22, no. 1, p. 6, Mar. 2024, doi: 10.1007/s10723-023-09727-1.

27. V. Varma, "Secure Cloud Computing with Machine Learning and Data Analytics for Business Optimization," J. Eng. Technol. Adv., vol. 4, no. 3, pp. 181–188, 2024, doi: 10.56472/25832646/JETA-V4I3P119.

28. Tanwar and D. P. K. K V, "Review on Machine Learning for Resource Usage Cost Optimization in Cloud Computing," Int. J. Res. Appl. Sci. Eng. Technol., vol. 11, no. 5, pp. 468–472, May 2023, doi: 10.22214/ijraset.2023.51489.

29. S. S. Sahoo, C. Thomas, A. Sheth, C. Henson, and W. S. York, "GLYDE-an expressive XML standard for the representation of glycan structure.," Carbohydr. Res., vol. 340, no. 18, pp. 2802–7, Dec. 2005, doi: 10.1016/j.carres.2005.09.019.

30. Mohammed Lateef et al., "Intelligent Cloud Resource Usage Potentially to Improve Task Scheduling by the use of Artificial Intelligence," F1000Research, vol. 15, p. 366, Sep. 2026, doi: 10.12688/f1000research.177203.3.

31. R. Aleem, S. Malik, A. Kashif, and U. B. Tayab, "Cloud computing resources utilization forecasting using machine learning," Eng. Res. Express, vol. 8, no. 7, p. 075217, Apr. 2026, doi: 10.1088/2631-8695/ae556f.

32. Javeed, A. Borg, H. Grahn, L. Lundberg, D. Patel, and S. Shirinbab, "A Two-Stage Forecasting System for CPU Workload Prediction in Private Clouds," Sep. 2026. doi: 10.48550/arXiv.2609.03457.

33. Han, "Optimization of Cloud Computing Resource Scheduling using Firefly Swarm Optimization Intelligence Algorithm," in 2025 3rd International Conference on Data Science and Information System (ICDSIS), IEEE, May 2025, pp. 1–6. doi: 10.1109/ICDSIS65355.2025.11070853.

34. R. Doukha and A. Ez-zahout, "Enhanced Virtual Machine Resource Optimization in Cloud Computing Using Real-Time Monitoring and Predictive Modeling," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 2, pp. 658–664, 2025, doi: 10.14569/IJACSA.2025.0160267.

35. Y. Wang and X. Yang, "Intelligent Resource

Allocation Optimization for Cloud Computing via Machine Learning," Adv. Comput. Signals Syst., vol. 9, no. 1, pp. 55–63, 2025, doi: 10.23977/acss.2025.090109.

36. Dixit, Himani, P. Bhardwaj, and R. Sachdeva, "Cloud-Driven AI: Enhancing Machine Learning Performance Through Distributed Computing," in 2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG), IEEE, Dec. 2024, pp. 1–5. doi: 10.1109/ICTBIG64922.2024.10911346.

37. P. Ramya, V. Ramya, and M. Babu Rao, "An Efficient E-Waste Management System through Energy-Aware Routing and Hybrid Optimization Deep Learning Routing on an IoT-Cloud Platform," in 2023 International Conference on Inventive Computation Technologies (ICICT), IEEE, Apr. 2023, pp. 102–107. doi: 10.1109/ICICT57646.2023.10134402.

38. Kumar, R. Goomer, and A. K. Singh, "Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters," Procedia Comput. Sci., vol. 125, pp. 676–682, 2018, doi: 10.1016/j.procs.2017.12.087.

39. V. Verma, "Improving Product Recommendations in Retail with Hybrid Collaborative Filtering and LSTM," Int. J. Eng. Sci. Math., vol. 10, no. 8, August, pp. 113–128, 2021.

40. Ullah, "Ai-Driven Optimization Of Resource Allocation And Cost Efficiency In Cloud Computing Environments," 2025.

41. H. Zhang, J. Li, and H. Yang, "Cloud computing load prediction method based on CNN-BiLSTM model under low-carbon background," Sci. Rep., vol. 14, no. 1, p. 18004, Aug. 2024, doi: 10.1038/s41598-024-68339-1.

42. H. Gandhi and A. Jain, "Cloud Cost Optimization Strategies Using Machine Learning Algorithms," Int. J. Innov. Sci. Res. Technol., vol. 9, no. 11, pp. 1004–1007, 2025, doi: 10.5281/zenodo.14831440.