ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 3 Issue 4 December 2023 / Page No: 166-175
Paper Id: JETA-V3I8P119 / Doi: 10.56472/25832646/JETA-V3I8P119

Original Article

Proactive Resource Utilization Prediction for Scalable Cloud Systems with Machine Learning

Siddhesh Amrale
Independent Researcher.

Received Date: 29 November 2023 Revised Date: 15 December 2023 Accepted Date: 28 December 2023

Abstract *— Cloud computing's unexpected and dynamic workload fluctuations, which have a substantial influence on system performance, operating costs, and user experience, make efficient and scalable resource management an ongoing problem. To address these challenges, this study proposes a proactive resource utilization prediction framework leveraging machine learning to enable adaptive, intelligent, and timely resource allocation in cloud environments. The framework utilizes the Microsoft Azure Traces 2017 dataset, which provides realistic telemetry data, to accurately forecast CPU usage trends. A comprehensive preprocessing pipeline—including data cleaning, feature engineering, and feature scaling—ensures high-quality input for model training and reliable predictive performance. Two proposed models, Linear Regression (LiR) and Bidirectional Long Short-Term Memory (Bi-LSTM), are employed to capture both linear patterns and complex temporal dependencies in data. For benchmarking, established comparison models—Autoregressive Neural Network, SVM, and VAR-GRU—are evaluated. Performance metrics include R^2, RMSE, MSE, and MAE. LiR achieved $R^2 = 0.9834$, RMSE = 0.0170, MSE = 0.00029, and MAE = 0.0125, while Bi-LSTM achieved $R^2 = 0.9733$, RMSE = 0.0217, MSE = 0.00047, and MAE = 0.0168, clearly outperforming the comparison models. Future work will focus on extending the framework to multi-resource forecasting, incorporating memory, network, and GPU metrics, and further enhancing cost-effectiveness and adaptive scaling in large-scale cloud environments.*

Keywords*—Resource Management, Machine Learning, Resource Utilization, Proactive Scaling, Resource Allocation, Cloud Computing.*

I. INTRODUCTION

The new digital infrastructure now requires cloud computing as a component, changing the way organizations and individuals retrieve, store, and manipulate data. It provides on-demand access to resources, flexibility and affordable solutions in different fields, such as scientific research, engineering and business processes [1][2]. The actual strength of cloud systems is scalability and elasticity since the resources may be expanded or reduced according to the demand[3]. However, there is the problem of efficient use of resource. Also, the dynamic, unpredictable, and heterogeneous nature of workloads grows, and the methods of allocating jobs to employees become increasingly less realistic, leading to inefficiency and performance bottlenecks [4].

The dynamic and heterogeneous nature of workloads in cloud deployments means that the requirements of the resources continuously fluctuate, and many of the changes are normally not predictable manually [5]. The ineffective allocation of resources occurs because control of this variability is done by inflexible rules or manual actions. The low resource utilization levels in the present-day data centers can impact the high availability levels of infrastructures that resulting in inefficiencies in operations and high costs of energy. A solution to these problems [6] is to proactively manage the resources. In the prospect of the future resource demands, optimization of performance, decrease of the energy wastage, and operational costs are possible, which leads to the stabilization of the services and cost efficiency.

The traditional or reactive approach to resource allocation, such as threshold-based monitoring and rule-based scaling, does not apply well when it comes to the modification of rapidly changing workloads. The over- or under-provisioning may lead to the system performance decline, breach of service-level agreements, and avoidable expenses and energy consumption, respectively. The existence of such problems points to one of the biggest shortcomings of current cloud management policies. The point is that it is hard to foresee the use of resources like memory, bandwidth, and CPU, and storage in a scalable manner [7][8]. An appropriate solution to this issue can help make systems in a large-scale cloud setup cost-efficient and reliable[9].

Cloud providers are not left behind as they strive to solve these problems using smart and automated resource management procedures. Being predictive, smart solutions forecast demand shifts based on historical and real-time usage information and assign resources to them [10][11]. This not only enhances performance and availability, this also brings about cost-efficient operations and reduced cost of operation. More precise forecasting mechanisms are also quite handy in improving improved scheduling, load balancing and service level assurance that in the long run, would translate to improved end-user quality of service[12][13].

The subject of machine learning (ML) has also become a viable issue that can be used to model nonlinear and complex behaviours of a system. ML also allows the proactive supply of resources instead of a response to historical utilization and present utilization data[14]. With this change, it is possible to predict the workload, the strategies plans of scaling down and minimized waste [15][16]. The ML approaches have already managed to predict the use of cloud resources by being successful in predictive autoscaling, anomaly detection and capacity planning and therefore it would best suit the use of the methods to predict the use of them. ML is deployed to make the providers more scaled, cost-effective and reliable leading to smart and data-driven cloud management.

A. Motivation and Contributions of the Study

The rapid evolution of cloud computing, service providers are facing more challenges of effectively handling the dynamic workloads within scalable infrastructures. Unexpected spikes in the use of resources may cause performance declines, higher operating costs and poor user experiences. The conventional methods of statical allocation of resources generally cannot respond to these changes in real-time, and, hence, either over-provision or under-provide resources. Another potentially useful solution is the machine learning-based predictive analytics because this tool enables making decisions proactively. The ability of the cloud systems to predict resource usage allows them to optimize the performance, lower expenses while upholding service level agreements (SLAs). This study is guided by the need to establish a smart, data-oriented system that has the potential to forecast resource needs and provide scalable, efficient, and dynamic resource management to the modern cloud-based environment. The current study adds the following aspects:

- They propose a robust methodology to forecast the use of resources in cloud systems which involves pre-processing of data, feature engineering of data, and multifaceted models of ML.
- Bidirectional Long Short-Memory (Bi-LSTM) and Linear Regression (LiR) are employed to identify both linear trends and complex time-related trends of the usage of resources to ensure a high degree of prediction accuracy.
- The framework is applied to the Microsoft Azure Traces 2017 data, which provides the real telemetry data, thus, the predictions in the dynamic cloud environment become more feasible.
- The models are thoroughly verified with the assistance of the different metrics (R^2, RMSE, MAE) and they present the full picture of predictive accuracy and reliability.
- The forecasts made by the framework can be used in pre-emptive scaling and provision of cloud services and reducing operation costs, preventing service outages, and increasing the overall efficiency of the systems.

B. Justification and Novelty of paper

To establish the sense of the necessity of adequate and effective management of cloud resources, the analysis suggests a dual-model architecture that combines Linear Regression (LiR) and to forecast both linear and multidimensional time series tendencies in cloud resource utilization, Bi-LSTM is used. The proposed models are compared to Autoregressive Neural Network, SVM, and VAR-GRU, compared to the current methods and demonstrate higher qualities of prediction. The real-world usage of the actual telemetry data would prove to be dependable and applicable in real-time in mobile cloud scenarios. A thorough analysis based on R^2, RMSE, MSE, and MAE emphasize accuracy, effectiveness, and strength to support the clever, flexible, and proactive use of resources and promote performance and cut down the costs of operation.

C. Organization of the paper

The paper is structured as follows: Section II examines relevant cloud resource management work. Section III presents the proposed methodology, detailing data pre-processing and the implementation of ML models. Section IV discusses the experimental configuration and the prediction models' performance outcomes. Finally, Section V concludes the study, highlighting key findings, practical implications, and potential directions for future research.

II. LITERATURE REVIEW

The evaluated research focus on CPU, GPU, and multi-resource workloads and explore several machine learning methods to forecast cloud resource utilization. Methods such as DCRNN, RNN, SARIMA-LSTM and SVR maximize forecasting, but they have scalability and real-time adaptation issues.

Al-Asaly et al. (2022) describe the challenges associated with the forecasting of future CPU resource demands of Software as a Service (SaaS) providers with changing workloads and the existence of multiple virtual machines (VMs). Their concept is an advanced DL technique that uses a diffusion convolutional recurrent neural network (DCRNN) to enhance workload prediction and provide cloud resources. This model's objective is to increase predicting accuracy under varying workload patterns and was tested on the real-world data of CPU usage on Planet Lab. It is found that results are greatly improved when compared to the current models, with the average absolute percentage inaccuracy was 0.18 and the root-mean-square error was 2.40 [17].

Cioca and Schuszter (2022) discuss the benefits of DL time-series analysis to use fewer computing resources, in the direction of greener data centres. They state that contemporary software development systems, such as container technology use like Kubernetes and Docker, tend to consume substantial resources, even when they are not active. The authors report a predictive system designed on the data and parameters of actual production systems in CERN that uses an RNN model that uses previous data to forecast future resource usage. This allows intelligent machine/container scaling down when it is not in use. Their results show possible carbon footprint savings in computing services, and CPU utilization improvement of up to 60-80% over classic approaches, which do not go deep in terms of historical analysis [18]

Anupama, Shivakumar and Nagaraja (2021) describe a new method of predicting workloads on clouds, which is directed at optimizing the use of resources through efficient resource administration plans. In order to manage workloads that are both seasonal and non-seasonal, the authors suggest a hybrid prediction model that blends machine learning and statistics. Seasonal data is estimated using the Seasonal Auto Regressive Integrated Moving Average (SARIMA) model, whereas non-seasonal workloads are estimated using the Autoregressive Integrated Moving Average (ARIMA) and LSTM networks depending on the results of the normality test. The predictive model predicts the number of resources needed in different periods and reveals that the LSTM model is much superior to ARIMA in forecasting irregular patterns and SARIMA in predicting future resource requirements. The study assists cloud service providers to balance workloads in order to avoid resource over or under provisioning [19].

Ntambu and Adeshina (2021) describe the major problems of security in cloud computing, especially because it incorporates multiple technologies, such as networks and virtualization. They also mention current security methods including intrusion detection and two-factor authentication, but they warn that there is a chance that hostile people might get access to virtual machines (VMs). The recommendations offered by the authors are proactive monitoring and a model of anomaly detection which operates on the Virtual machine (VM) resources and applies the ML algorithms to the work of One-Class Support Vector Machine and Isolation Forest (OCSVM).   The average F1-scores for hourly and daily time series are 0.97 and 0.89, respectively, indicating that OCSVM performed better than IFS in time series categorization [20].

Yeung et al. (2020) Discuss the importance of considering the utilization of GPUs in DL applications to optimize resource utilization and cloud cost-benefit evaluation. The current approaches of measuring GPU usage focus on online profiling of a single isolated device, i.e., a single task, and need separate profiling, which results in underutilization and reduced service capacity. In this regard, the authors suggest a prediction engine, which can predict the use of the GPUs in terms of multiple DL workloads without requiring a vast amount of online profiling. They make these predictions by extracting the data from the model computation graph. The results indicate that the prediction engine's Root Mean Squared Logarithmic Error (RMSLE) is 0.154, allowing DL schedulers to optimize the use of the GPU clusters by a maximum of 61.5 [21].

Abdullah et al. (2020) In order to reduce waste, highlight how cloud computing uses resources. Current methods use resource allocation and virtual machine consolidation, among other resource management systems. The significant degree of unpredictability in cloud resource utilization necessitates efficient prediction techniques. This study proposed the Support Vector Regression Technique (SVRT), A technique to supervised statistical learning for estimating the utilization of multi-attribute host resources. Training was conducted using the Sequential Minimal Optimization Algorithm using the Radial Basis Function kernel, the technique is found to be meaningfully better on prediction error (4-16% better) and training error (8-60% better) on real-world workloads of BitBrain, PlanetLab, and Google Cluster Workload Traces [22].

The Table I shows previous studies of workload prediction, which identify the methods, datasets, and gaps in multi-resource forecasting, scalability in real time, handling heterogeneous workloads, and sustainable optimization.

Table 1: Existing Studies on machine learning-based workload prediction or cloud resource utilization

References	Dataset	ML Approach	Key Outcomes	Limitations	Future Directions
Al-Asaly et al. (2022)	PlanetLab CPU usage traces	DCRNN – captures temporal and VM correlations	MAPE = 0.18, RMSE = 2.40; improved forecasting accuracy	Single resource (CPU) only; limited real-time testing; VM correlations not fully modeled	Extend to multi-resource forecasting; integrate adaptive real-time prediction; test under heterogeneous workloads
Cioca and Schuszter (2022)	CERN production software systems	RNN – predicts CPU and container usage	60%-80% reduction in computing power; reduced carbon footprint	Focused mainly on container scaling; limited generalization to other cloud setups	Generalize model to multi-resource prediction; test across diverse cloud platforms; incorporate dynamic workload patterns
Anupama, Shivakumar and Nagaraja (2021)	Simulated seasonal & non-seasonal workloads	Hybrid SARIMA + LSTM/ARIMA	SARIMA accurate for seasonal workloads; LSTM outperformed ARIMA for irregular workloads	Single resource only; hybrid selection requires normality tests; may not scale to large systems	Automate model selection; extend to multi-resource prediction; validate in large-scale cloud environments

Ntambu and Adeshina (2021)	Sampled VM workload traces	OCSVM / Isolation Forest – anomaly detection	High F1-scores (OCSVM: 0.97 hourly, 0.89 daily)	Focused on anomaly detection, not proactive prediction; limited to CPU	Integrate anomaly detection with predictive resource allocation; extend to multi-resource, real-time monitoring
Yeung et al. (2020)	GPU workloads of DL models	DL Graph Prediction – predicts GPU usage	RMSLE = 0.154; improved GPU cluster utilization up to 61.5%	Only GPU considered; requires workload graph info; does not include CPU/memory	Develop multi-resource prediction; enable prediction without detailed workload graph; support heterogeneous cloud workloads
Abdullah et al., (2020)	BitBrain, PlanetLab, Google Cluster Workload Traces	SVR (RBF kernel) – predicts multi-resource usage	Accuracy improved 4%-16%; error reduced 8%-60%	Offline evaluation; scalability issues for large datasets; temporal correlations not fully captured	Apply deep learning/hybrid approaches for real-time multi-resource prediction; enhance scalability; model temporal dependencies

III. METHODOLOGY

The proposed methodology focuses on building an efficient predictive framework for resource utilization in scalable cloud systems, as presented in Figure 1. To guarantee data consistency, quality, and appropriateness for model training, it starts with data preparation, which includes crucial procedures including data cleaning, feature engineering, and feature scaling. The Microsoft Azure Traces 2017 dataset is then utilized, containing real-world telemetry data including CPU usage, memory, and network traffic, which are key indicators of system performance. In order to assess the model's performance, the data is first preprocessed, after which it is partitioned into one group of 90% training data and the other group of 10% testing data. Two prediction models that maintain linear trends and time-dependent correlations in data include Linear Regression (LiR) and Bidirectional Long Short-Term Memory (Bi-LSTM). The model's performance is measured on the basis of R^2, RMSE, MSE and MAE, which are good measures of the performance of the models. The final products contribute to enhanced resource management and are proactive in dynamic computing in the clouds.

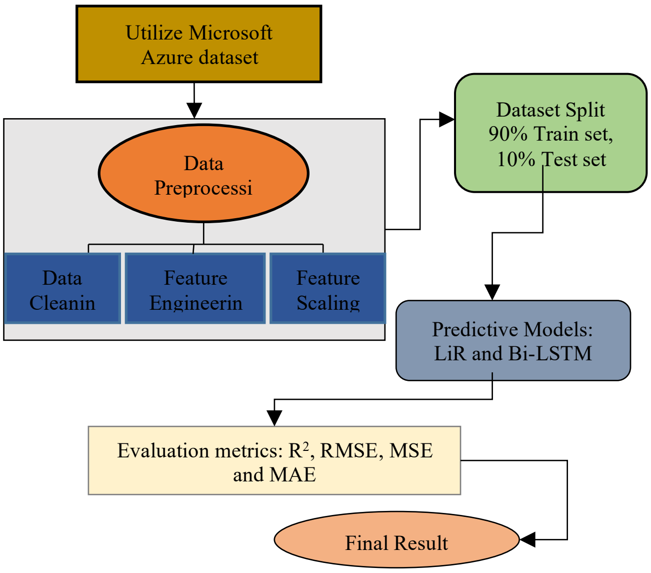
Utilize Microsoft Azure dataset
Data Preprocessi
Data Cleanin
Feature Engineerin
Feature Scaling
Dataset Split 90% Train set, 10% Test set
Predictive Models: LiR and Bi-LSTM
Evaluation metrics: R2, RMSE, MSE and MAE
Final Result

Figure 1: Proposed Predictive Framework for Resource Utilization in Scalable Cloud Systems

A. Data Collection

The Microsoft Azure Traces 2017 data is an example of cloud telemetry data trace of real-life data collected in the Azure data centres, which entails major measurements like CPU, network traffic, memory and disk I/O data. It provides the time series data about the performance of the virtual machine, and one can utilize the findings to create workload prediction and resource optimization forecasting models. This research primarily relies on the data of CPU usage to forecast the trends of CPU usage and network transmission. This information facilitates the ideal dynamic cloud modelling behavior that increases the effectiveness and performance of large-scale cloud computing systems.

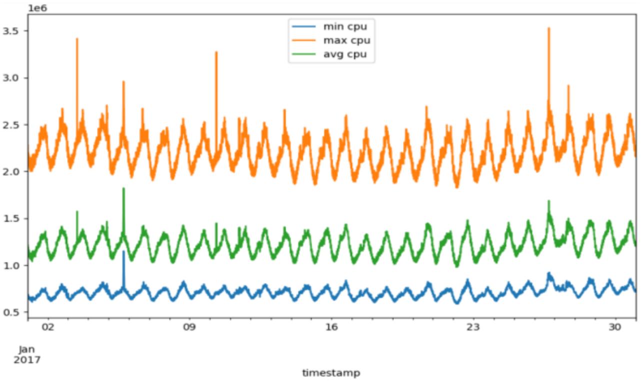

Figure 2: CPU Utilization Trends in the Dataset

Figure 2 shows patterns of CPU utilization of the Microsoft Azure Traces, which were captured in January 2017. It displays the CPU utilization as time passes in three different ways: minimal (blue), maximum (orange), and average (green). The x-axis is the time, whereas the y-axis is the CPU utilization in millions. The data shows periodic changes that can be observed with obvious peaks and troughs that indicate repetitive workload patterns common in cloud-based operations. The maximum CPU usage displays higher spikes, indicating varying demand intensity, while the average and minimum values remain relatively stable, reflecting consistent background activity and efficient resource management within the Azure infrastructure.

B. Data Preprocessing

The preprocessing of data is an important process in the process of data analysis and ML that requires cleaning, transforming, and structuring raw data into an appropriate format. It guarantees the quality of data, consistency, and model training preparation.

C. Data Cleaning

Data cleaning is an important pre-processing task that is designed should improve the dataset's consistency, quality, and dependability before developing a model [23]. The procedure of cleaning is provided below:

- Treat missing data to ensure that data is not lost and also prevent the risk of bias in the model.
- Identify and eliminate outliers that can violate model performance.
- Eliminate the inconsistency or duplication of entries to maintain integrity of the data.
- Reformat data and format to have consistency in each feature.

D. Feature Engineering

The feature engineering is a critical process where raw data are selected and transformed into useful inputs that enhance better model performance and prediction accuracy. CPU usage, network throughput and timestamp are critical attributes that are taken into consideration in this research to ensure important patterns in resource use are captured. In the case of categorical variables, they are converted to appropriate numerical data to make them compatible with ML and DL models. Additional time characteristics are obtained to enhance trend analysis that ultimately leads to improved learning and forecasting of resource requirements of the model.

E. Feature Scaling

The technique of normalizing feature scaling refers to the variety of characteristics or independent variables in a dataset by means of data preparation. It makes sure that the attributes play an equal role in model training, and features with bigger magnitudes do not take over smaller ones [24]. Minmax normalization is one such method, and it normalizes the information to a certain range, often [0,1]. As displayed by Equation (1), the normalized data sample x', can be obtained using the original data sample x:

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})} \times \left(x_{new\_max} - x_{new\_min}\right) + x_{new\_min} \tag{1}$$

F. Data Splitting

In a 90:10 ratio, the dataset is divided into training and testing subsets, with 90% going towards model training and 10% towards model testing, in order to precisely evaluate the model's generalization potential, accuracy, and performance.

G. Proposed Linear Regression Model

A statistical method called linear regression is used to describe the connection between one or a criterion or response variables (also called a dependent variable) plus additional predictor variables (sometimes called independent variables). The following Equation (2) represents the model.

$$y = b_0 + b_1 * x \tag{2}$$

where x is the predictor value, b_0 is the line's slope, b_1 is the intercept, and indicates the fitted or estimated value [25]. The anticipated value of $\hat{y}$ when the predictor assumes a value of x=0 is known as the intercept, whereas the slope may be used to calculate how much the predictor affects the dependent variable for each unit of predictor change. The least squares approach is used to estimate the coefficients b_0 and b_1 in regression, which minimises the number of squares that the difference between actual and predicted values deviates from. They are calculated by the following Equations (2&4):

$$b_0 = \bar{y} - b_1 * \bar{x} \tag{3}$$

$$b_1 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2} \tag{4}$$

In which n represents the observations, the average of the predictor variable is denoted as $\bar{x}$ and the average of the response variable is denoted as $\bar{y}$. LR is a simple but effective method of understanding associations and estimation, which is the foundation of most sophisticated prediction modelling methods.

H. Proposed BiLSTM Model

The two parallel LSTMs that make up Bidirectional Long Short-Term Memory (BiLSTM) process data in an anticlockwise and clockwise manner, respectively. The latent state of BiLSTM, which is the sum of the two before and after states, has the potential to hide the present and future states at any given moment [26]. LSTM usually consists of three gates (forget, input, and output) at any given sequence time t. In this investigation, the state memory unit $Ct{-}1$ and the hidden state h_{t-1} of the end sequence were added to the forgetting gate together with the current input vector x_t. The forget gate output, f_t, is produced using a sigmoid activation function; the calculation Equation (5) is:

$$f_t = \sigma\left(W_f h_{t-1} + U_f x_t + b_f\right) \tag{5}$$

where the weights are represented by W_f, Uf, and the bias by b_f. There are two components to the input gate: The sigmoid activation function is utilised in the first component [27], and its outcome is i_t. The tanh function is used in the second portion, and the result is a_t. When both elements are combined, it is determined which vector must be maintained in the state memory unit. The formula for calculation is shown in Equation (6-7):

$$i_t = \sigma\left(W_f h_{t-1} + U_i x_i + b_i\right) \tag{6}$$

$$a_t = tanh(W_a h_{t-1} + U_a x_i + b_a) \tag{7}$$

The first, $Ct{-}1$, is the product of the forget gate's output ft, and the second, the product of the input gate's outputs i_t and a_t.The Equation (8) are given below:

$$C_t = C_{t-1} \odot f_t + i_t \odot a_t \tag{8}$$

The Hadamard product is represented by $\odot$. There are two components to the concealed state h_t update. Equation (9&10) expresses the tanh activation function and the hidden state C_t, which constitute the second section.

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{9}$$

$$h_t = o_t \tanh \odot (C_t) \tag{10}$$

This enables the prediction of the final two LSTMs' output, the subsequent temporal output. In Figure 3, the BiLSTM structure is displayed.

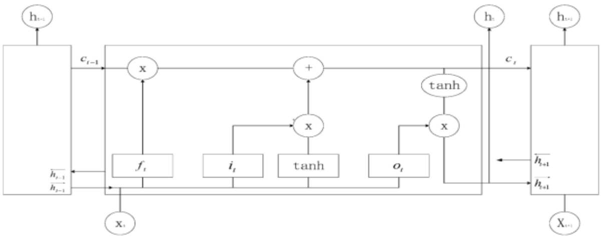

Figure 3: Schematic Diagram of Bilstm Structure [28]

I. Performance Metrics

In order to assess the process's quality and the predictability of the outcomes, assessing the prediction model's performance is crucial. A number of performance metrics are frequently employed to assess the model's precision and predictive ability. These measures are determined using the following formulas:

1) Mean Squared Error (MSE)

The MSE is a way to find the average of the squares of the errors [29]. In this scenario, the difference between the estimated and actual amounts is the error. It represents the estimator's quality measure and is sometimes referred to as the risk function. Equation (11) expresses the MSE function.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2 \tag{11}$$

where y_i is the actual value, n is the quantity of observations and the anticipated value, p_i. The function's return value is always non-negative and preferably greater than 0.

2) Root Mean Squared Error (RMSE)

The RMSE is the standard deviation of the residuals, which is the difference between the actual and predicted values. It displays the degree to which the data and the best-fit line correspond. Equation (12) provides an expression for the RMSE function.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2} \tag{12}$$

3) R-squared

R2 is the coefficient of determination, often referred to as the goodness function [30], This evaluates the regression's accuracy in predicting the real data points. Equation (13), which expresses the R2 function.

$$R^2 = 1 - \frac{\sum_i (y_i - p_i)^2}{\sum_i (y_i - A)^2} \tag{13}$$

where A is the mean of the observed or anticipated data. Perfect match to the data is indicated by a value nearer 1.

4) Mean Absolute Error (MAE)

The mean absolute error (MAE) is the difference between the estimated and actual values. There is never a value that is less than zero. Equation (14) shows how to find the MAE:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - y_i| \tag{14}$$

A thorough assessment of prediction models is made possible by these indicators taken together, allowing for objective comparison and selection of the most effective approach for a given dataset.

IV. RESULT ANALYSIS AND DISCUSSION

The proposed study's foundation utilizes ML to investigate the proactive use of resources in scalable cloud systems. The two models, Linear Regression and BiLSTM, were coded in Python using scikit-learn (LR) and TensorFlow/Keras (BiLSTM), and training and testing were done in Google Colab. Table II provides a comparative analysis of their performance measures, where LR has better R2 (0.9834) and fewer errors (RMSE: 0.0170, MSE: 0.0003, MAE: 0.0129) than BiLSTM (R2: 0.9733, RMSE: 0.0217, MSE: 0.0005, MAE: 0.0170). These outcomes suggest that Linear Regression is more accurate and efficient to compute, and should thus be used as an alternative in proactive management of cloud resources, and to aid in optimum scalability in the dynamic cloud environments.

Table 2: Performance comparison of proposed linear regression and BILSTM models for resource utilization prediction

Metrics	Linear Regression	BiLSTM
R2 Score	0.9834	0.9733
RMSE	0.0170	0.0217
MSE	0.0003	0.0005
MAE	0.0129	0.0170

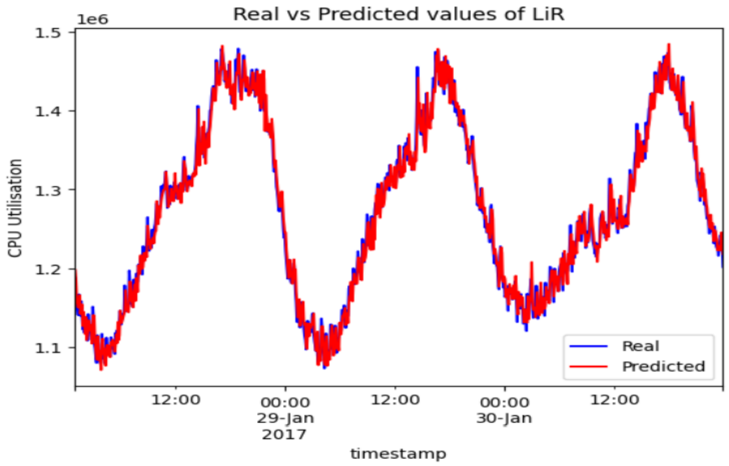

Figure 4. Comparison of Real vs. Predicted CPU Utilization using Linear Regression Over Time

Figure 4 illustrates the performance of a predictive model for CPU Utilization over a period spanning approximately two and a half days, from late January 28 to early January 31, 2017. The actual (real) CPU utilization levels are shown by the blue line, measured on the y-axis, while the model's expected values are displayed by the red line. The graph displays a clear diurnal pattern in CPU usage, with peaks occurring around midday and minimums near midnight each day. More importantly, the predicted values (red line) follow the actual ones (blue line) in the observed timestamps, which signifies the substantial efficiency of the model to account for both the small variations in resource utilization and the high-level periodic patterns.

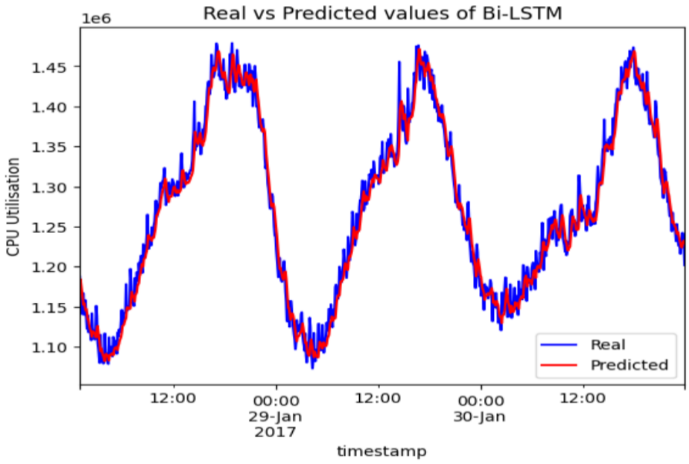

Figure 5. Comparison of Real vs. Predicted CPU Utilization using Bi-LSTM Over Time

Figure 5 demonstrates the high-performance rate of predicting CPU utilization using a neural network model for Bi-LSTM. The blue line (Real) during the timeframe between late January 28 and early January 31, 2017, indicates the actual CPU usage with a very high level of diurnal periodicity, with a high level of utilization at midday. The real values are well tracked by the red line (Predicted), both in the cyclical ups and downs, and at the levels of most of the smaller fluctuations. The near correspondence between the actual and projected numbers indicates the Bi-LSTM model's high fidelity in modelling the complex temporal dependencies and patterns in resource utilization, making it a robust predictor for operational planning and resource management.

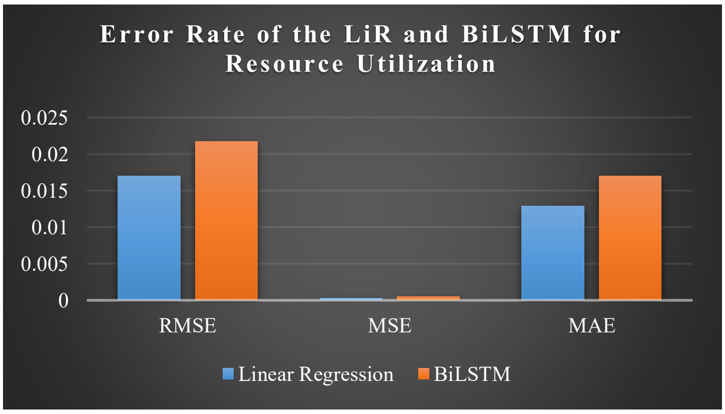

Figure 6: Error Rates Comparison of LiR and Bi-LSTM Models

The performance of the two prediction models, Bi-LSTM and Linear Regression (LiR), is contrasted in Figure 6. Three standard error measurements serve as the basis for the comparison MAE, MSE, and RMSE. The chart shows that for RMSE and MAE, the Bi-LSTM (orange bars) exhibits a higher error rate (approximately 0.022 and 0.017, respectively) than Linear Regression (blue bars) (approximately 0.017 and 0.013, respectively). Conversely, for the MSE metric, both models show very low and nearly equal error rates, close to zero. The overall visualization suggests that, based on RMSE and MAE, the simpler Linear Regression model performed slightly better than the complex Bi-LSTM in predicting resource utilization.

A. Comparative Analysis & Discussion

This section presents a performance comparison of the proposed models, Linear Regression and BiLSTM, for predicting resource utilization in scalable cloud systems. Their results are summarized in Table III with performance that is compared to the current approaches like Autoregressive Neural Network (RMSE: 0.61449), SVM (0.504) and VAR-GRU (0.3295). These two proposed models are much better than these more complex and traditional approaches and achieve a significantly smaller value of 0.0170 of RMSE when using Linear Regression and 0.217 when using BiLSTM. Linear Regression is more accurate and faster to calculate as compared to BiLSTM. These results suggest that the proposed models can be highly helpful to make relevant and valid predictions that are capable of actively managing resources and factors that make cloud-based systems most efficient in the context of scalability.

Table 3: Performance comparison of proposed models and existing approaches for resource utilization prediction

Model	RMSE
Autoregressive Neural Network[31]	0.61449
SVM[32]	0.504
VAR-GRU[33]	0.3295
Linear Regression	0.0170
Bi-LSTM	0.0217

The results demonstrate that the proposed models, Linear Regression (LiR) and Bi-LSTM, are capable of predicting cloud resource usage with slightly better accuracy and increasing computational efficiency; however, Bi-LSTM can adhere to complex time variations. The suggested models' superiority over Autoregressive Neural Network, SVM, and VAR-GRU suggests their high robustness and dependability. The results demonstration that the opportunity to dynamically allocate resources to minimize over-allocation and operating expenses is provided by active prediction. The LiR is a lightweight framework that enables quick predictions, while Bi-LSTM provides long dependencies, making the framework adaptable to various large-scale cloud deployments.

V. CONCLUSION AND FUTURE SCOPE

This is needed to get a correct forecast on how the cloud resources utilized to provide maximum performance, low price and reliability of the system. Two models were developed based on them, and they are: Linear Regression (LiR) and Bi-LSTM, that was used to model both linear and temporal patterns of workload. The comparison was done with an Autoregressive neural network, SVM and VAR-GRU, where the measures of evaluation are R^2, RMSE, MSE, and MAE. LiR achieved $R^2 = 0.9834$, RMSE = 0.0170, MSE = 0.00029, and MAE = 0.0125, while Bi-LSTM achieved $R^2 = 0.9733$, RMSE = 0.0217, MSE = 0.00047, and MAE = 0.0168, outperforming the comparison models. LiR is a lightweight and computationally efficient model, and Bi-LSTM is successful at capturing the time dynamics, and the framework is capable of scaling to real-time. This form of predictive solution facilitates proactive allocation of resources, minimizes over-providing of resources as well and enhances the dynamic cloud environments' cost-effectiveness. However, in the future, prediction will be extended to other resources, such as memory, network, and GPU, to allow managing multiple resources holistically. Further improvements will consider adaptive ensemble procedures that incorporate both statistical and DL algorithms and incorporate reinforcement learning to make completely autonomous scaling choices. Scalability, resilience, and practical applicability is going to be tested in different types of production-quality cloud structures and deployed in real time in order to provide a solid baseline of intelligent, self-adaptive cloud resource management and optimization.

VI. REFERENCES

[1] Y. C. Lee and A. Y. Zomaya, “Energy efficient utilization of resources in cloud computing systems,” *J. Supercomput.*, vol. 60, no. 2, pp. 268–280, 2012.

[2] D. Buchaca, J. Ll. Berral, C. Wang, and A. Youssef, “Proactive container auto-scaling for cloud native machine learning services,” in *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, 2020, pp. 475–479.

[3] S. Srinivasan, R. Sundaram, K. Narukulla, S. Thangavel, and S. B. Venkata Naga, “Cloud-Native Microservices Architectures: Performance, Security, and Cost Optimization Strategies,” *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. 4, no. 1, pp. 16–24, 2023, doi: 10.63282/3050-9246.IJETCSIT-V4I1P103.

[4] B. R. Cherukuri, “Quantum machine learning: Transforming cloud-based AI solutions,” *Int. J. Sci. Res. Arch.*, vol. 1, no. 1, pp. 110–122, 2020, doi: 10.30574/ijsra.2020.1.1.0041.

[5] T. Mehmood, S. Latif, and S. Malik, “Prediction of Cloud Computing Resource Utilization,” in *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT and IoT, HONET-ICT 2018*, 2018. doi: 10.1109/HONET.2018.8551339.

[6] V. M. L. G. Nerella, “A Database-Centric CSPM Framework for Securing Mission-Critical Cloud Workloads,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 209–217, 2022.

[7] R. Tandon and D. Patel, “Evolution of Microservices Patterns for Designing HyperScalable Cloud-Native Architectures,” *ESP J. Eng. Technol. Adv.*, vol. 1, no. 1, pp. 288–297, 2021, doi: 10.56472/25832646/JETA-V1I1P131.

[8] A. Kushwaha, P. Pathak, and S. Gupta, “Review of optimize load balancing algorithms in cloud,” *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, pp. 1–9, 2016.

[9] A. P. and S. Pandya, “Compliance-Driven Data Governance: A Survey on GDPR, and HIPAA in Cloud Databases,” *Int. J. Curr. Eng. Technol.*, vol. 12, no. 6, pp. 828–836, 2022, doi: https://doi.org/10.14741/ijcet/v.12.6.18.

[10] V. K. Singh, “Lessons Learned from Large-Scale Oracle Fusion Cloud Data Migrations,” *Int. J. Sci. Res.*, vol. 10, no. 10, pp. 1662–1666, Oct. 2021, doi: 10.21275/SR21101083620.

[11] G. Modalavalasa and S. Pillai, “Exploring Azure Security Center : A Review of Challenges and Opportunities in Cloud Security,” *ESP J. Eng. Technol. Adv.*, vol. 2, no. 2, 2022, doi: 10.56472/25832646/JETA-V2I2P120.

[12] V. Verma, “Big Data and Cloud Databases Revolutionizing Business Intelligence,” *TIJER – Int. Res. J.*, vol. 9, no. 1, 2022.

[13] V. M. L. G. Nerella, “Automated cross-platform database migration and high availability implementation,” *Turkish J. Comput. Math. Educ.*, vol. 9, no. 2, pp. 823–835, 2018.

[14] S. Garg, “AI/ML driven proactive performance monitoring, resource allocation and effective cost management in saas operations,” *Int. J. Core Eng. Manag.*, vol. 6, no. 6, 2019, [Online]. Available: https://www.ssrn.com/abstract=5267257

[15] D. R. Avresky, P. Di Sanzo, A. Pellegrini, B. Ciciani, and L. Forte, “Proactive scalability and management of resources in hybrid clouds via machine learning,” in *2015 IEEE 14th International Symposium on Network Computing and Applications*, 2015, pp. 114–119.

[16] M. P. Yadav, Rohit, and D. K. Yadav, “Resource provisioning through machine learning in cloud services,” *Arab. J. Sci. Eng.*, vol. 47, no. 2, pp. 1483–1505, 2022.

[17] M. S. Al-Asaly, M. A. Bencherif, A. Alsanad, and M. M. Hassan, “A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment,” *Neural Comput. Appl.*, 2022, doi: 10.1007/s00521-021-06665-5.

[18] M. Cioca and I. C. Schuszter, “A System for Sustainable Usage of Computing Resources Leveraging Deep Learning Predictions,” *Appl. Sci.*, vol. 12, no. 17, 2022, doi: 10.3390/app12178411.

[19] Anupama, K. C, Shivakumar, B. R, Nagaraja, and R, “Resource Utilization Prediction in Cloud Computing using Hybrid Model,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 4, 2021, doi: 10.14569/IJACSA.2021.0120447.

[20] P. Ntambu and S. A. Adeshina, “Machine Learning-Based Anomalies Detection in Cloud Virtual Machine Resource Usage,” in *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, 2021, pp. 1–6. doi: 10.1109/ICMEAS52683.2021.9692308.

[21] G. Yeung, D. Borowiec, A. Friday, R. Harper, and P. Garraghan, “Towards {GPU}utilization prediction for cloud deep learning,” in *12th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 20)*, 2020.

[22] L. Abdullah, H. Li, S. Al-Jamali, A. Al-Badwi, and C. Ruan, “Predicting Multi-Attribute Host Resource Utilization Using Support Vector Regression Technique,” *IEEE Access*, vol. 8, pp. 66048–66067, 2020, doi: 10.1109/ACCESS.2020.2984056.

[23] N. Marie-Magdelaine and T. Ahmed, “Proactive autoscaling for cloud-native applications using machine learning,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, 2020, pp. 1–7.

[24] P. J. M. Ali, “Investigating the Impact of min-max data normalization on the regression performance of K-nearest neighbor with different similarity measurements,” *ARO-The Sci. J. Koya Univ.*, vol. 10, no. 1, pp. 85–91, 2022.

[25] M. Daraghmeh, S. B. Melhem, A. Agarwal, N. Goel, and M. Zaman, “Linear and logistic regression based monitoring for resource management in cloud networks,” in *2018 IEEE 6th international conference on future internet of things and cloud (FiCloud)*, 2018, pp. 259–266.

[26] J. Peter, “Improving the Auto scaling mechanism in Cloud computing environment using Support Vector regression and Bi-LSTM,” Dublin, National College of Ireland, 2022.

[27] N.-M. Dang-Quang and M. Yoo, “An efficient multivariate autoscaling framework using bi-lstm for cloud computing,” *Appl. Sci.*, vol. 12, no. 7, p. 3523, 2022.

[28] Y. Zhang, Y. Liu, X. Guo, Z. Liu, X. Zhang, and K. Liang, “A BiLSTM-Based DDoS Attack Detection Method for Edge Computing,” *Energies*, vol. 15, no. 21, 2022, doi: 10.3390/en15217882.

[29] P. Nehra and A. Nagaraju, “Host utilization prediction using hybrid kernel based support vector regression in cloud data centers,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, Part B, pp. 6481–6490, 2022, doi: https://doi.org/10.1016/j.jksuci.2021.04.011.

[30] V. Chudasama and M. Bhavsar, “A dynamic prediction for elastic resource allocation in hybrid cloud environment,” *Scalable Comput.*, vol. 21, no. 4, pp. 661–672, 2020, doi: 10.12694:/scpe.v21i4.1805.

[31] Q. Zia Ullah, S. Hassan, and G. M. Khan, “Adaptive Resource Utilization Prediction System for Infrastructure as a Service Cloud,” *Comput. Intell. Neurosci.*, vol. 2017, 2017, doi: 10.1155/2017/4873459.

[32] S. Banerjee, S. Roy, and S. Khatua, “Efficient resource utilization using multi-step-ahead workload prediction technique in cloud.,” *J. Supercomput.*, vol. 77, no. 9, 2021.

[33] S. Ouhame, Y. Hadi, and A. Ullah, “An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model,” *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10043–10055, 2021.