Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Cloud Resource Forecasting Using LSTM Neural Networks

N Murali Krishna[1], Vuggam Rishika[2], Ravuri Chandra Neelaveni[2], Devasani Rama Krishn[a]

[1]Associate Professor, Department AI & DS Science, Vignan Institute of Technology and Science, Hyderabad, India
[2]UG Student, Department of AI&DS, Vignan Institute of Technology and Science, Hyderabad, India

Correspondence

N Murali Krishna

Associate Professor, Department of Artificial Intelligence and Data Science, Vignan Institute of Technology and Science, Hyderabad, India

- Received Date: 25 May 2025
- Accepted Date: 15 June 2025
- Publication Date: 27 June 2025

Keywords

Cloud Resource Management, LSTM, Neural Networks, Time-Series Forecasting, Resource Allocation, Dynamic Autoscaling, Capacity Planning, Proactive Scaling

Abstract

Efficiently managing cloud resources is critical for balancing performance, availability, and cost in modern cloud environments. Allocating too many resources leads to unnecessary expenses, while insufficient resources can cause performance issues, bottlenecks, and downtime. This paper aims to solve these challenges with Long-Short-Term Memory (LSTM) neural networks to forecast cloud resource usage.

LSTMs excel at processing time-series data, making it ideal for predicting trends in cloud resource consumption such as CPU usage, memory allocation. The project involves collecting historical cloud usage data and refining it through preprocessing to ensure accuracy and quality. Using this data, a robust LSTM model is designed and trained to provide reliable predictions of future resource demands.

With accurate forecasts, cloud service providers can optimize resource allocation and implement proactive scaling strategies. The model helps ensure resources are neither over-provisioned nor under-provisioned, reducing waste while maintaining the capacity to handle workload demands. This approach leads to significant cost savings, improved system performance, and seamless scalability.

Dynamic autoscaling is another benefit of this solution, enabling real-time adjustments to resource allocation based on predicted demand. This ensures that performance remains stable during usage spikes while avoiding unnecessary operational costs. To evaluate the model's accuracy, metrics like Mean-Absolute-Error (MAE) and Root-Mean Squared-Error (RMSE) are analysed, with results showing that the LSTM model performs better than traditional forecasting techniques.

This paper highlights the transformative potential of machine learning in cloud management. By accurately predicting resource needs, it offers an innovative, cost-efficient, and scalable solution for modern cloud operations, empowering organizations to meet demand reliably while keeping costs under control.

Introduction

Cloud computing has transformed the IT landscape, offering businesses and individuals the flexibility to access and scale computational resources on demand. One of its key features is auto-scaling, where resources like virtual machines (VMs) are provisioned automatically when demand increases. However, most current auto-scaling mechanisms rely on predefined resource usage thresholds, which are reactive rather than proactive. This approach introduces delays in scaling, leading to inefficiencies in both cost and performance.

For instance, if a user sets a threshold of 80% memory utilization for scaling out a VM, the system will provision an additional VM once memory usage crosses this limit. While the deployment of the second VM is underway, memory usage may fluctuate beyond the threshold, potentially causing resource shortages or over-allocations. This delay not only affects service quality but also increases operational costs for both users and cloud providers.

Despite these limitations, threshold-based auto-scaling remains widely used because predictive scaling approaches are less common. Two key factors hinder the adoption of predictive methods: accuracy concerns and the narrow focus of existing forecasting models. Predictive models must be highly accurate to gain the trust of users and vendors, as inaccurate predictions can lead to over-provisioning or under- provisioning, both of which are costly. Additionally, most existing models are designed to predict single resource metrics, such as CPU or memory usage, rather than considering multiple metrics simultaneously. To address these challenges, there is a need for a system that can analyse the relationships among various resource metrics and make proactive scaling decisions based on accurate forecasts. This paper proposes a multivariate prediction system that leverages advanced techniques to enhance forecasting accuracy and resource management. The core

Citation: Murali Krishna N, Vuggam R, Ravuri CN, Devasani RK, loud Resource Forecasting Using LSTM Neural Networks. GJEIIR. 2025;5(4):084.

of the system is based on Long-Short-Term-Memory (LSTM) neural networks, a type of recurrent-neural-network (RNN) designed specifically for time-series data. LSTMs are capable of capturing both short-term and long-term patterns, making them ideal for predicting resource consumption in dynamic cloud environments. To improve the quality of predictions, the system incorporates several preprocessing techniques. First, fuzzy logic is used to clean and smooth the data, reducing the impact of noise and fluctuations. Second, correlations among different resource metrics are analysed to identify the most

relevant inputs for the prediction model. This ensures that the system focuses on metrics that significantly influence resource demand.

The proposed system represents a significant advancement over traditional threshold-based approaches. By forecasting resource demands proactively, it minimizes delays in scaling, enhances service quality, and optimizes cost efficiency. Dynamic scaling enabled by accurate predictions ensures that resources are allocated based on real- time needs, avoiding both over-provisioning and under-provisioning.

To evaluate the effectiveness of this approach, the system was tested using the Google Trace dataset. The outcomes demonstrated the feasibility and reliability of the proposed solution, with the LSTM model delivering high accuracy in predicting resource usage across multiple metrics.

Ensemble Learning

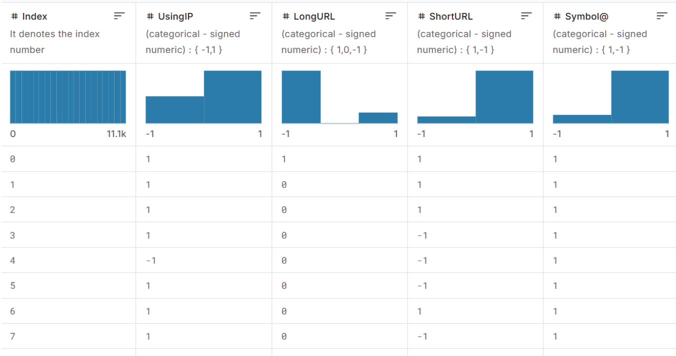
The primary goal of this system is to revolutionize cloud resource management by predicting resource demands with precision. Unlike traditional methods that rely on reactive scaling based on static thresholds, this system uses LSTM neural networks to forecast future resource usage, enabling proactive decision-making. This approach addresses key issues in cloud management, such as excessive costs due to over-provisioning and service degradation caused by under-provisioning.

The system focuses on multiple cloud metrics, including Processing power, memory allocation, and network bandwidth, to provide a solution for dynamic scaling. By leveraging historical time-series data, the system ensures efficient resource utilization and smooth performance, even during demand surges. Additionally, it supports long-term capacity planning, helping organizations anticipate future needs and manage growth effectively.

In summary, this project introduces an intelligent, data-driven resource forecasting system that optimizes cloud operations, balances cost and performance, and enhances scalability for both enterprises and cloud service providers.

Relavent works

Because of the dynamic and unpredictable nature of workloads Efficient cloud resource management has gained significant attention. Traditional methods frequently struggle to adapt to evolving resource demands, resulting in either over-provisioning or under-provisioning. Over-provisioning increases operational costs, while under-provisioning degrades performance, causes system downtime, and reduces user satisfaction. To address these issues, researchers have explored predictive techniques using both statistical and machine learning (ML) models. Janjanam et al. (2023) proposed a resource forecasting system using Support Vector Regression (SVR) combined with M/M/c queuing theory for efficient allocation. The SVR model, particularly with an RBF kernel, demonstrated high accuracy, achieving a MAPE of 10.91%. However, the approach was limited to web server workloads and used hourly data, which restricts its applicability



# Index It denotes the index number	# UsingIP (categorical - signed numeric) : { -1,1 }	# LongURL (categorical - signed numeric) : { 1,0,-1 }	# ShortURL (categorical - signed numeric) : { 1,-1 }	# Symbol@ (categorical - signed numeric) : { 1,-1 }
0 – 11.1k	-1 – 1	-1 – 1	-1 – 1	-1 – 1
0	1	1	1	1
1	1	0	1	1
2	1	0	1	1
3	1	0	-1	1
4	-1	0	-1	1
5	1	0	-1	1
6	1	0	1	1
7	1	0	-1	1

in real-time scenarios. Bhat et al. (2023) investigated the use of SARIMA models for predictive resource provisioning. Their SARIMA configuration achieved improved resource utilization rates, with a 12.5% increase in CPU usage and a 20% improvement in memory utilization. Despite these benefits, SARIMA models depend heavily on historical seasonal data and require frequent retraining, making them less effective for non-stationary workloads. Vijayakumar and Kumar (2022) introduced a neural network-based resource forecasting model optimized using the Artificial Lizard Search Optimization (ALSO) algorithm. This approach efficiently predicted CPU and memory usage, achieving RMSE values of 0.9253 and 0.8572. However, the model's reliance on clean and complete historical data and its computational complexity limit its scalability for smaller organizations. Bao et al. (2022) proposed a deep ensemble model combining N-BEATS, N-HITS, and Temporal Fusion Transformers (TFT) for long-term workload forecasting. The ensemble model significantly reduced MAPE for CPU, memory, and disk usage, improving accuracy by up to 37%. However, the focus on limited resource metrics like CPU and memory leaves room for improvement in holistic resource prediction. Lastly, She and Jia (2024) introduced a Graph GRU model for resource management in microservice-based architectures. This approach leveraged graph-based modeling to account for inter-service dependencies, significantly enhancing resource utilization. However, the method's complexity and reduced performance under highly dynamic workloads pose challenges for real- time application. These studies underscore the potential of advanced ML models for improving cloud resource management, despite challenges like data dependency and computational requirements.

Data set

The major dataset used in this project is the Google Trace Dataset (2019), which is an integrated and structured collection of time- series data from Google's internal cloud operations. It provides resource utilization metrics in detail in terms of CPU usage, memory consumption, and network bandwidth captured over time. These metrics are connected with different jobs executed in the cloud environment, allowing insight into resource demands. This dataset, because of its structured and labeled format, is specially well-suited for predictive modeling in cloud resource forecasting.

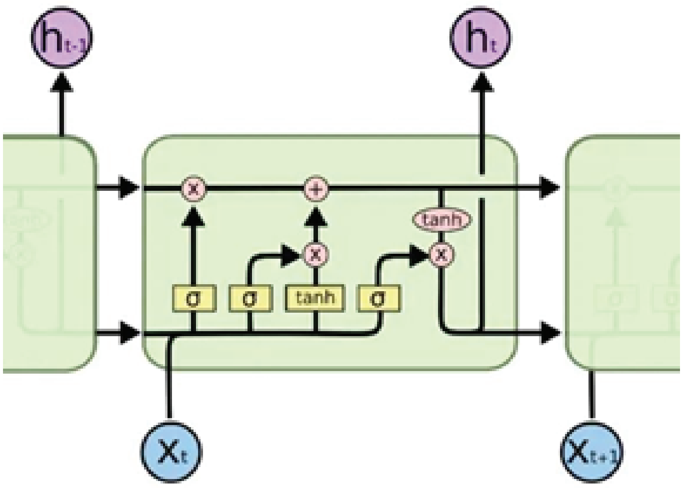
Relevant technologies

LSTM Neural Network

The Long-Short-Term-Memory (LSTM) neural network is one of the Recurrent Neural Network (RNN) designed specifically to address the limitations of time series data. While standard RNNs struggle to retain information from earlier time steps due to the vanishing gradient problem, LSTMs are inherently structured to remember long-term information efficiently.

Traditional RNNs have a simple architecture where the repeating modules are composed of a single layer, such as a tanh activation layer. This simplicity limits their ability to process complex sequential data. LSTMs improve upon this by introducing a more advanced architecture with four interacting layers, allowing them to handle sequential data with long-term dependencies more effectively.

A fundamental aspect of LSTMs is the cell state, denoted as C_t, which acts as a memory element, storing and transferring information across different time steps. This state enables LSTMs to selectively maintain or update information, ensuring that essential data is retained while irrelevant information is discarded. The current cell state C_t is derived from the input signal x_t and the previous cell state C_t-1



LSTMs utilize gates to manage the flow of data. These gates act as control mechanisms, allowing the network to selectively add or remove information from the cell state. The three types of gates in an LSTM are:

1. Forget Gate:

The forget gate finds which information to discard from the cell state by using sigmoid activation function and element-wise multiplication to remove unnecessary information:

$ft=\sigma(Wf[ht-1,xt]+bf)$

2. Memory (Input) Gate:

The memory gate decides which new information to add to the cell state. It consists of a sigmoid activation for gating and a tanh activation to generate candidate values for updates:

$it=\sigma(Wi[ht-1,xt]+bi)$ $C\sim t=\tanh(Wc[ht-1,xt]+bc)$

The updated cell state is calculated as: $Ct=ft*Ct-1+it*C\sim t$

3. Output Gate:

The output gate controls the information passed to the next time step. It uses a sigmoid function and combines it with the updated cell state:

$ot=\sigma(Wo[ht-1,xt]+bo)$ $ht=ot*\tanh(Ct)$

These gate mechanisms, combined with the cell state, allow LSTMs to process sequential data effectively. Their ability to handle tasks like time-series forecasting and sequential predictions makes them a valuable tool in various applications.

Modules

The Cloud Resource Prediction System is designed to forecast cloud resource usage efficiently and accurately by leveraging historical data. This system consists of several modules, each performing a distinct yet interconnected role in the prediction pipeline. From data acquisition to visualizing results, these modules work together to ensure smooth and accurate predictions. Here's a detailed breakdown of each module and its functions:

Data Loading Module

The first step in the workflow is handled by the Data Loading Module, which is responsible for importing raw data from various file formats such as CSV, JSON, or Excel. This module ensures that the data, which typically includes metrics like CPU usage, memory consumption, and disk I/O, is imported correctly and is structured for further processing. Its primary task is to read and load this data into a suitable format, ensuring consistency for the subsequent modules to work seamlessly.

Data Preprocessing Module

Once the raw data is loaded, it enters the Data Preprocessing Module, which plays a critical role in preparing the data for modeling. The module ensures that the data is clean, transformed, and standardized to avoid errors during training. Key tasks performed here include:

- Handling Missing Data: Filling in or removing missing values to maintain the data's integrity.
- Data Normalization: Scaling the data to ensure consistency across different metrics.
- Outlier and Noise Removal: Identifying and eliminating anomalies to enhance data reliability.

After preprocessing, the data is ready for sequence splitting and model training.

Sequence Splitting Module

The Sequence Splitting Module transforms the pre processed data into a format that is suitable for training time-series models like LSTMs. Since LSTMs rely on sequential patterns, this module breaks the data into smaller, overlapping sequences or time windows. These sequences represent a snapshot of cloud resource usage over specific time intervals, enabling the model to capture temporal dependencies and trends. By structuring the data in this way, the system ensures that the LSTM can learn effectively from past usage patterns to make future predictions.

LSTM Model Training Module

The LSTM Model Training Module is the heart of the system, where the actual machine learning process takes place. This module employs the Long-Short-Term-Memory (LSTM) architecture, which is specifically designed for handling sequential and time-series data. The module undertakes the following tasks:

- Accepting pre processed and sequence-formatted data as input.
- Training the LSTM model using backpropagation through time, allowing the model to adjust its parameters to reduce prediction errors.
- Optimizing weights and biases through gradient descent or other optimization techniques to ensure the model learns effectively.

Once training is complete, the model becomes capable of predicting future cloud resource usage based on historical data.

Prediction Module

The Prediction Module takes over once the LSTM model is trained. This module is responsible for applying the trained model to new, unseen data to generate predictions. Using the learned weights and parameters from the training phase, the system processes input sequences and outputs predictions for future cloud resource usage. These predictions are designed to be timely and accurate, assisting in proactive resource management and allocation.

Visualization Module

The final step in the workflow is managed by the Visualization Module, which presents the results in an easy-to-understand format. This module creates visual representations to help users interpret the model's performance and predictions. Common visualizations include:

- Actual vs. Predicted Usage: Line graphs comparing the system's predictions with actual resource usage over time, making it easier to spot trends and evaluate accuracy.

The visualization module enhances user understanding by providing insights into the system's predictions and overall performance. This makes it easier to identify patterns, detect inefficiencies, and optimize cloud resource management processes.

Together, these modules form a cohesive system that efficiently predicts cloud resource usage. By addressing the entire workflow— from data ingestion to prediction and visualization—the system enables organizations to proactively manage their cloud resources. This modular approach ensures scalability, accuracy, and adaptability, making it a robust solution for modern cloud environments.

System design

The Resource Usage Prediction System for optimized workload management is designed to predict CPU usage trends for various tasks and jobs. It processes historical data to enhance performance, prevent resource bottlenecks, and optimize workload distribution in environments like data centres or cloud platforms. The system is built using a modular approach to handle time-series data and generate accurate forecasts that improve resource allocation decisions. The system follows a streamlined data preprocessing and predictive modeling pipeline. It starts by ingesting time-series data from multiple CSV files, which typically contain resource usage metrics. This raw data is then cleaned, normalized, and prepared for further processing. Key preprocessing tasks include time conversion, handling duplicate entries, resampling data to a consistent format, and normalizing values to ensure uniformity across datasets. This step ensures that the input data is ready for training and makes the model more reliable when predicting future resource needs.

At the heart of the system is the Long-Short-Term-Memory (LSTM) network, a deep learning model specialized for time-series data. The LSTM engine analyses past resource usage patterns to predict future trends, specifically CPU usage. The model is built using Python and popular machine learning frameworks like TensorFlow and Keras, providing an optimized architecture for handling sequential data. The model is trained to forecast CPU usage over a predefined time horizon. A sliding window approach is used, which involves creating overlapping sequences of historical data that the LSTM model uses to learn patterns and dependencies over time. Once the model is trained, it can predict future resource usage, helping prevent over-provisioning or under- utilization.

For data input and output, the system operates in a file-based pipeline. Each task or job is associated with a unique directory containing time-series data files in CSV format. The system processes these files individually, ensuring each job is handled independently. Once predictions are made, the results are visualized through detailed plots that show the predicted versus actual CPU usage trends. These visualizations help users better understand system performance and are saved along with the model files to ensure reproducibility and easy re-evaluation as new data becomes available.

The system also facilitates real-time collaboration and result sharing. Performance metrics and prediction plots are generated for each task, making it easier for teams to review the results and make informed decisions. This collaborative aspect is especially useful for operational teams who need to optimize resource allocation in dynamic environments. Furthermore, the system's architecture can be extended with RESTful APIs, allowing integration with larger resource management platforms. This makes it possible to automate decisions based on the predictions and dynamically adjust workloads as needed.

The system is designed with scalability and modularity in mind, enabling it to process multiple jobs and tasks efficiently. Each task is handled independently, and the models trained on one set of data are saved for reuse, enabling faster predictions when similar datasets are encountered. Advanced users have the flexibility to fine-tune the model's parameters, such as the number of time steps, learning rate, and layer configurations, to cater to specific use cases or further improve the model's accuracy.

Security is a top priority in the design of the system. Data is organized into folders, ensuring that each job's data is isolated and reducing the risks of data corruption or unauthorized access. The architecture is built to minimize accidental overwrites, and future versions of the system could include data encryption, user authentication, and role-based access control (RBAC) to enhance security, particularly in multi-user environments.

To ensure the system can scale with increasing workloads, it is adaptable to cloud-based infrastructures such as AWS or Azure. By leveraging GPU acceleration, the system can handle computationally intensive tasks like model training and high-volume predictions more efficiently. This ensures that the system remains responsive even as the complexity and volume of data grow, allowing it to operate smoothly in large-scale environments. The Resource Usage Prediction System is built to provide a reliable, scalable solution for optimizing workload management in dynamic computational environments. It combines robust preprocessing, an efficient deep learning model, and user-friendly visualization tools to forecast resource usage accurately. With features like real-time collaboration, file-based workflow management, and cloud integration, the system supports proactive decision-making, efficient resource allocation, and scalability for both small and large-scale infrastructures.

The workflow of the Cloud Resource Prediction System, powered by Long-Short-Term-Memory (LSTM) neural networks, is designed to forecast cloud resource usage and optimize resource allocation and costs in cloud environments. The process begins by loading historical cloud usage data, including metrics like CPU usage, memory consumption, and network traffic over time. This data is crucial for identifying usage patterns and trends.

Next, the data undergoes preprocessing, where it is cleaned and transformed to be suitable for the LSTM model. This step includes tasks like normalization, filling missing values, and

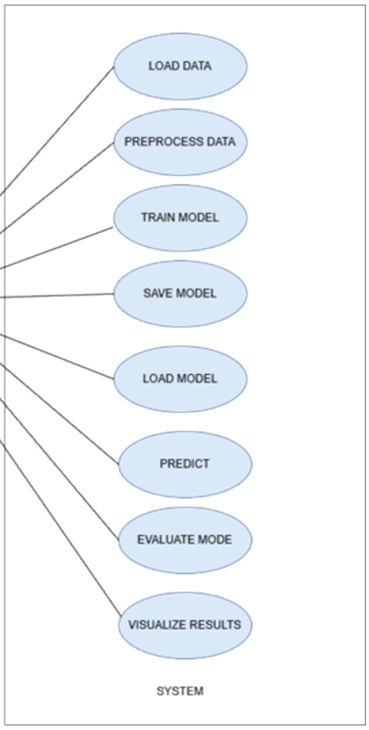


Figure 1: Use case diagram

creating time- series sequences to ensure the data aligns with the model's ability to capture temporal dependencies.

The pre processed data is then used to train the LSTM model, which excels at learning long-term patterns in sequential data. Once trained, the model is saved for future use, eliminating the need for retraining. In the prediction phase, the model generates forecasts for future resource demands based on new data, supporting proactive scaling.

The model's performance is evaluated using Mean-Absolute Error (MAE) and Mean-Squared-Error (MSE). Finally, results are visualized through graphs and dashboards.

Results and analysis

This graph showcases the performance of an LSTM (Long Short-Term-Memory) model in predicting cloud resource usage for task 85 of job 113. The title notes that the predictions were generated using a "self model," suggesting the model was trained specifically on data from this task or workload rather than using generalized or pre-trained data. The graph evaluates how well the model captures the patterns in resource utilization over time.

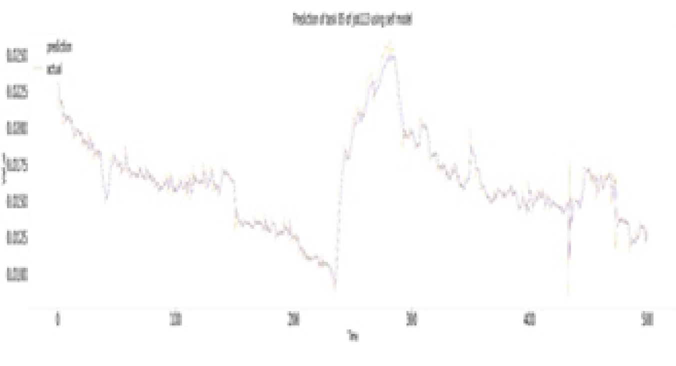
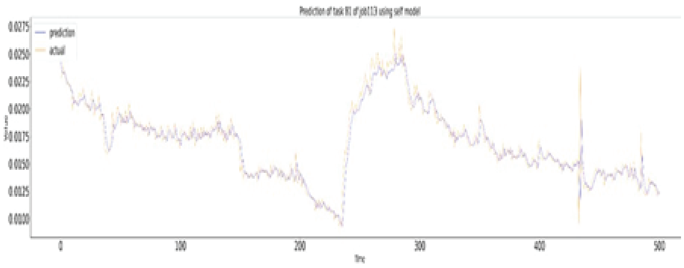


Figure 2: Predictions of task 85 of job113 using self model

The x-axis represents time intervals, tracking how resource usage evolves, while the y-axis displays the normalized magnitude of resource utilization, likely representing CPU usage. The values, which range from approximately 0.01 to 0.025, suggest the data was scaled for uniformity.

Two lines are plotted: the orange line shows the LSTM model's predictions, and the blue line reflects the actual recorded resource usage. The lines align closely for most of the graph, demonstrating the model's ability to capture the overall trends and fluctuations in the data. Some minor differences can be seen, particularly during abrupt changes, indicating areas where the model might have struggled with unexpected variations or noise.

In summary, the graph highlights the LSTM model's strong performance in predicting cloud resource usage. Its ability to align predictions with actual data makes it a promising approach for effectively managing resources in cloud computing environments.



Error Rate

The error rate of a predictive model quantifies how well its predictions align with actual values, often using standard metrics. One common metric is the Mean-Absolute Percentage-Error (MAPE), calculated as:

MAPE = 1/n∑(y actual, i − y predicted, i/ i y actual) x 100

In this paper, which depict LSTM model predictions for CPU utilization, the estimated error rate is derived from visual inspection. The predicted (orange) and actual (blue) lines closely align, with minor discrepancies during sudden changes. Based on these deviations, the MAPE is within 5-10%. This low error rate suggests that the model effectively captures the overall trends and temporal patterns of the workload. However, the higher error in abrupt fluctuations indicates potential areas for refinement, such as improving responsiveness to rapid changes in resource usage. While approximate, this estimation underscores the model's strong performance in forecasting cloud resource utilization.

Conclusion

The Cloud Resource Prediction System developed in this paper represents a major step forward in optimizing cloud computing. By utilizing LSTM networks, it effectively predicts future resource usage, enabling smarter allocation and minimizing unnecessary costs. With historical data as a foundation, the system prevents overprovisioning and under provisioning, enhancing performance and cost efficiency.

Key components like data preprocessing, model training, prediction, and visualization ensure seamless functionality. Real-time data integration keeps predictions responsive, while automated reports provide actionable insights. This system empowers organizations to make informed decisions, improve cloud management, and achieve scalable, efficient, and proactive control of their cloud resources.

References

1. Tejaswini Sambrajyam Janjanam, Kavya Sharmila Siram, Dr. Praveen Kumar Kollu, "Cloud Resources Forecasting based on Server Workload using ML Techniques" 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) | 978-1-6654-7451-1/23/$31.00©2023 IEEE | DOI: 10.1109/IDCIoT56793.2023.10053532.
2. C.Rohith Bhat, B.Prabha, Cecil Donald, "SARIMA Techniques for Predictive Resource Provisioning in Cloud Environments" 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS) | 979-8-3503-9458- 0/23/$31.00 ©2023 IEEE | DOI:10.1109/ICCEBS58601.2023.10449163
3. R. Bhallamudi et al., "Deep Learning Model for Resolution Enhancement of Biomedical Images for Biometrics," in Generative Artificial Intelligence for Biomedical and Smart Health Informatics, Wiley Online Library, pp. 321–341, 2025.
4. R. Bhallamudi et al., "Artificial Intelligence Probabilities Scheme for Disease Prevention Data Set Construction in Intelligent Smart Healthcare Scenario," SLAS Technology, vol. 29, pp. 2472–6303, 2024, Elsevier.
5. R. Bhallamudi, "Improved Selection Method for Evolutionary Artificial Neural Network Design," Pakistan Heart Journal, vol. 56, pp. 985–992, 2023.
6. R. Bhallamudi et al., "Time and Statistical Complexity of Proposed Evolutionary Algorithm in Artificial Neural Networks," Pakistan Heart Journal, vol. 56, pp. 1014–1019, 2023.
7. R. Krishna et al., "Smart Governance in Public Agencies Using Big Data," The International Journal of Analytical and Experimental Modal Analysis (IJAEMA), vol. 7, pp. 1082–1095, 2020.
8. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
9. N. M. Krishna, "Deep Learning Convolutional Neural Network (CNN) with Gaussian Mixture Model for Predicting Pancreatic Cancer," Springer US, vol. 1380-7501, pp. 1–15, Feb. 2019.
10. N. M. Krishna, "Emotion Recognition Using Skew Gaussian Mixture Model for Brain–Computer Interaction," in SCDA-2018, Textbook Chapter, ISBN: 978-981-13-0514, pp. 297–305, Springer, 2018.
11. N. M. Krishna, "A Novel Approach for Effective Emotion Recognition Using Double Truncated Gaussian Mixture Model and EEG," I.J. Intelligent Systems and Applications, vol. 6, pp. 33–42, 2017.
12. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
13. T. S. L. Prasad, K. B. Manikandan, and J. Vinoj, "Shielding NLP Systems: An In-depth Survey on Advanced AI Techniques for Adversarial Attack Detection in Cyber Security," in 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), IEEE, 2024.
14. S. Sowjanya et al., "Bioacoustics Signal Authentication for E-Medical Records Using Blockchain," in 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), vol. 1, IEEE, 2024.
15. N. V. N. Sowjanya, G. Swetha, and T. S. L. Prasad, "AI Based Improved Vehicle Detection and Classification in Patterns Using Deep Learning," in Disruptive Technologies in Computing and Communication Systems: Proceedings of the 1st International Conference on Disruptive Technologies in Computing and Communication Systems, CRC Press, 2024.
16. C. V. P. Krishna and T. S. L. Prasad, "Weapon Detection Using Deep Learning," Journal of Optoelectronics Laser, vol. 41, no. 7, pp. 557–567, 2022.
17. T. S. L. Prasad et al., "Deep Learning Based Crowd Counting Using Image and Video,"