International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 3, Issue 1, 143-151, 2022
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P115

Original Article

Intelligent Predictive System of Cloud Resource Utilization Forecasting with Advanced Deep Learning

Uday Kumar Ragireddy[1], Prasanth Varma Addepalli[2], Sridhar Reddy Bandaru[3], Dhuli Shyam[4], Prabu Manoharan[5], Muzaffer Hussain Syed[6]
[1]Sr Technical Program Manager Vdrive IT Solutions, Inc, Richardson, Texas.
[2]Data Engineer II Cox Automotive Corp Svcs LLC, Atlanta, Georgia.
[3]Program Management, IT ,Microsoft, Senior ACE Engineer Redmond, WA.
[4]Business Application, IT, Nagase Holdings America Corp, Manager, Application & Software Development NYC, NY.
[5]Information Technology, Bourns Inc, HRIS Manager, California, USA.
[6]Sr Software Developer, Visual Technologies, Plano, TX.

Abstract - *Time series forecasting is important in cloud data centers to efficiently allocate resources in the form of cloud resources, since forecasting demand allows effective use of computing resources and reduces costs. Traditional methods based on conventional machine learning or statistical analysis often fall short at capturing complex temporal patterns, leading to low prediction accuracy and inefficient resource use. Three models are proposed in this research. Using actual Microsoft Azure trace data, CNN, GRU, and a hybrid CNN-GRU are used to forecast resource usage. The experimental results show that the CNN-GRU model achieves the highest performance, with the lowest error rates (MSE: 0.0002, MAE: 0.0136, RMSE: 0.0164) and the maximum R2 of 98.23%, compared to the single CNN (R2: 94.59%) and GRU (R2: 97.45%). The proposed models show much better predictive accuracy than traditional forecasting methods such as CP-SAE and LSTM. The findings validate the importance of combining spatial and temporal learning features for powerful forecasting of cloud resources. The paper provides an actionable model for proactive resource distribution and lays the groundwork for future developments based on multi-source real-time data and an advanced architecture to further increase prediction accuracy in dynamic cloud-based environments.*

Keywords - *Cloud Computing, Resource Utilization Prediction, CPU Usage Forecasting, Machine Learning, Deep Learning, Cloud Resource Management.*

1. Introduction

Cloud computing (CC) provides an adaptable architecture that enables applications to obtain the resources they need before executing essential applications on virtual machines [1]. Cloud service companies frequently employ pay-as-you-go pricing to offer clients flexibility and reduce expenses. A significant increase in cloud users and the development of cloud-based apps to access various cloud computing services are a result of the broad range of developments in CC technology. CC services are used in many scientific applications, resulting in a range of cloud resource utilization [2]. To satisfy fluctuating user demand, effective resource management is therefore necessary. In cloud computing environments, efficient resource management can maximize resource utilization, reduce costs, and enhance performance [3]. Effective resource management is achieved by forecasting resource use.

Predicting resource usage has been thoroughly researched, and there is a wealth of literature accessible [4][5]. Several factors affect resource prediction models, including accuracy, the model's time and memory complexity, managing many resources, etc. Accurately predicting resource consumption is challenging due to several factors, including network speed, disk I/O, CPU and memory usage, etc. There may be connections between, for instance, CPU and memory consumption and disk I/O and memory [6]. Determining and forecasting relationships among different types of resources is challenging. Consequently, the forecast results not be useful in real-world situations. The auto-scaler based on cloud resource prediction needs to assess multiple indicators simultaneously to make appropriate scaling decisions [7].

The application of ML techniques in research has received more attention in recent years [8]. Many resource indicators are handled concurrently by the prediction approach. ML uses load-balancing cloud topologies, workload complexity, and services to predict customer requests and burden [9]. Workload cloud architectures enhance resource performance and allocate them in accordance with demand, while service load balancing in cloud architectures increases utilization, improves availability, and enhances performance by using a deep learning model that forecasts load based on requirements [10]. The framework uses several techniques to improve model predictions and maximize cloud resource utilization [11][12]. Proactive scaling, improved load balancing, and enhanced cloud resource utilization are enabled by ML and DL algorithms that concurrently model these intricate interactions and forecast resource demand. Thus, the following might be used to summarize this work's primary contributions:

- A robust pre-processing pipeline using real Azure traces with cleaning, feature engineering, and min–max normalization for practical and accurate resource prediction.
- Development of a hybrid CNN-GRU model that successfully identifies temporal and spatial trends in data on cloud resource usage.
- Demonstration of improved forecasting accuracy through extensive evaluation using MAE, MSE, RMSE, and R^2 metrics.
- offering a reliable, broadly applicable framework for predicting cloud resource use that minimizes overfitting and improves scalability.

The research is important because it enables more effective prediction of cloud resource utilization, which can be used to improve resource allocation and system performance. It is unique in that it overcomes the limitations of single models and traditional forecasting techniques by leveraging a hybrid CNN-GRU model that combines temporal dependency learning and spatial feature extraction. This methodology along with a configured pre-processing pipeline offers a high-performing and scalable platform to real-time, data-intensive cloud resource management.

1.1. Structure of Paper

This paper's remaining sections are organized as follows: a description of related work is found in Section II; Section III focuses on the proposed methodology; Section IV concentrates on the experiments performed; and Section V presents conclusions and further work.

2. Literature Review

The literature contains several methods for predicting resource usage based on both single- and multi-ML techniques, some of which are listed below. Kumar and Singh (2019) propose a differential evolution-based workload prediction method for efficient VM distribution. The forecast accuracy of the proposed strategy is evaluated using Google's real-world trial and compared with the most sophisticated prediction methods. Saw a notable decrease in forecast inaccuracy of up to 71% and 88%, respectively, in comparison with forecasting techniques based on back propagation and linear regression [13].

Wang et al. (2019) create and implement a tool that enables users to choose models and independently determine the optimal parameter combination. Users may simply develop an elastic, offline resource forecast system that offers resources using RPT, an openly improvable toolkit. The system comprises many scalable prediction modules. These include dataset import, model selection, parameter adjustments, and prediction outcomes. The model can be widely used to forecast future cloud resource use and to select appropriate assessment criteria to evaluate its effectiveness [14].

Li et al. (2019) developed a neuromorphic system based on cogent confabulation using historical record data across all dimensions to predict resource consumption. The system simulates a probability network that is carefully adjusted to support the prediction application, leveraging correlations across observations across several dimensions. The experimental data suggest that the cogent confabulation model-based VM resource utilization prediction has an inherent advantage for dynamic prediction with a larger prediction window and achieves greater accuracy than prior work. Using accurate confabulation-based virtual machine resource prediction, cloud resource management can boost energy efficiency (in terms of electricity prices) by up to 26.52% [15].

Shaw, Howley, and Barrett (2018). This paper presents a novel predictive, anti-correlated VM placement method and compares the most popular prediction models using actual workload traces. In comparison with some of the most popular placement regulations, Empirical data show that the suggested technique reduces energy consumption by 18% and service violations by over 47% [16].

Hassan, Chen and Liu (2018) presented DL is the foundation of DEARS, an Elastic and Automatic Resource Scheduling system for cloud apps. Based on historical workload, the LSTM model is applied to predict future request demand, offering a proactive and reactive strategy. The restriction assessment, VM provisioning, and dynamic consolidation modules each handle VM allocation independently. The performance of resource allocation is then improved reactively by iteratively applying the SLA's feedback. Experiments using real-world data demonstrate the methodology's viability and effectiveness. The high prediction accuracy facilitates a more appropriate allocation. Furthermore, compared to the baselines, a superior server-side trade-off between quality of service and SLAs is obtained. [17].

Lyu et al. (2017) offer a practical method for balancing economic profitability and high availability. Three more modules the forecast, adjustment, and collection modules were used to create a forecast approach. To increase forecast accuracy, the forecast module employs several ML algorithms. Carried out a comparative study, and the findings demonstrate the efficacy of the suggested forecasting method [18].

Although there have been significant advancements in predicting resource utilization, current methods lack real-time flexibility, cross-platform performance, and scalability for highly dynamic cloud workloads. As noted in Table I, earlier research is primarily concerned with increasing accuracy in controlled settings. It does not strongly support heterogeneous resource use, workload variation, or multi-objective optimization. Thus, it can be concluded that more adaptive and robust prediction framework incorporating real-time data, hybrid learning models, and proactive scheduling is clearly needed to increase reliability and decision-making in the contemporary cloud

Table 1: Comparison of Recent Studies on Resource Utilization Prediction Techniques

Author	Technique / Model Used	Contribution	Findings	Future Study
Kumar & Singh (2019)	Differential Evolution-based workload prediction	Developed an evolutionary forecasting scheme for accurate VM workload prediction using Google trace data	Reduced forecast error by 71% vs. backpropagation and 88% vs. linear regression	Extend to multi-cloud environments; integrate with real-time auto-scalers
Wang et al. (2019)	RPT – Resource Prediction Toolbox (ML model selection & tuning)	Built a modular, extensible prediction toolbox including dataset import, model selection, parameter tuning, and output evaluation	Provides flexible and scalable forecasting for cloud environments	Enhance toolbox with deep learning modules and real-time prediction support
Li et al. (2019)	Neuromorphic Cogent Confabulation Model	Designed multidimensional neuromorphic prediction model exploiting correlations in historical data	Improved prediction accuracy; increased energy efficiency by 26.52%	Explore hybrid neuromorphic–DL models; expand prediction window adaptability
Shaw, Howley & Barrett (2018)	ML model comparison + Anti-correlated VM placement	Introduced a predictive placement strategy using anti-correlated VM allocation	Reduced energy usage by 18% and service violations by 47%	Integrate with dynamic orchestration; test with diverse real-world workloads
Hassan, Chen & Liu (2018)	DEARS Framework (LSTM-based prediction + SLA-driven scheduling)	Proposed proactive and reactive resource scheduling using LSTM for workload prediction	Improved prediction accuracy; better quality of service–SLA trade-off and efficient VM allocation	Extend to hybrid cloud, containerized workloads, and multi-layer SLAs
Lyu et al. (2017)	Multi-module ML-based forecasting (Forecast, Adjustment, Collection modules)	Designed a forecasting mechanism using multiple ML techniques to enhance accuracy	Verified efficiency and improved performance across cloud scenarios	Incorporate reinforcement learning and adaptive forecasting in dynamic cloud setups

3. Methodology

The research analyzes Microsoft Azure trace data, which undergoes cleaning, verification, feature engineering and min-max normalization. A hybrid CNN-GRU model is intended to examine temporal relationships with GRU units and spatial patterns with CNN layers. The data is divided into subgroups for testing, validation, and training. The model is trained on processed data, and its ability to forecast resource use is assessed using the MAE, MSE, RMSE, and R2 metrics.

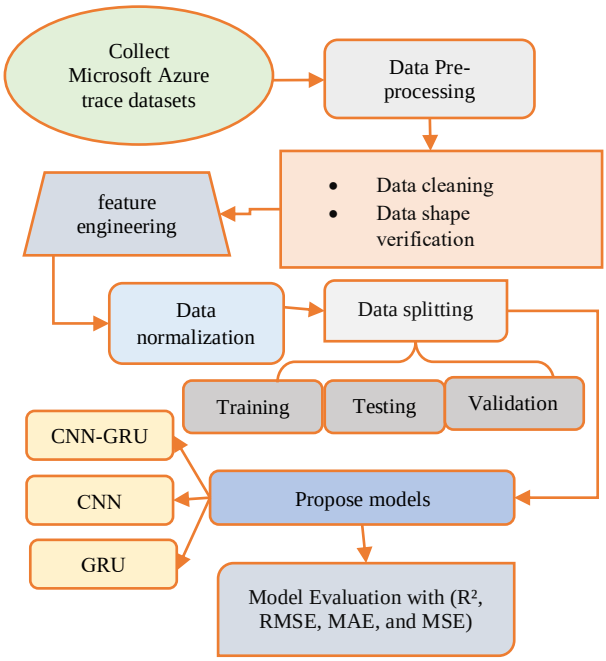

Fig 1: Proposed Flowchart For Resource Utilisation Prediction

Additionally, Figure 1 illustrates the Research Methodology for Cloud Computing Resource Utilization Prediction.

3.1. Data Acquisition and Pre-processing

Microsoft Azure traces from GitHub offer CPU usage data for forecasting trends in network transmission performance and CPU utilization. Key resource measurements, such as CPU utilization, which are essential to the study's predictive models, are available in the databases. The data was cleaned, feature-extracted and normalized. These steps are listed below:

- Data Cleaning: Clean up the datasets to get rid of inconsistencies, outliers, and missing numbers.
- Data Shape Verification: The dataset shape was confirmed, ensuring the correct number of rows and columns remained after filtering.

3.2. Feature Engineering

The feature engineering was done by deriving important temporal and system-level features, such as CPU usage and time-of-day effects, to model cyclical and load-sensitive behaviour. Categorical variables were appropriately coded, and numerical features were normalized or standardized as needed to ensure consistent scaling and improve model convergence.

3.3. Normalization

Each resource in the datasets is normalized independently, and its maximum capacity in relation to all machines in the trail is represented by a value of 1. The calculation for this procedure is displayed in Equation (1):

$$x_m = \frac{x - x_{min}}{x_{max} - x_{min}} \times (high - low) + low \qquad (1)$$

Where x is the value before the feature value is processed, x_{min} is the minimum of all the original features, and x_{max} Is the maximum of all the original features. The mapping interval's maximum and minimum values are denoted by high and low, respectively.

3.4. Data Splitting

Splitting the datasets in a 70:30:10 ratio into training, validation, and test sets.

3.5. Propose Hybrid CNN-GRU Model

The CNN-GRU hybrid model was created for cloud resource utilization. The rationale for this approach is based on the similarities between the two models' advantages—GRUs are better at learning temporal dependencies in sequential network data. At the same time, CNNs are better at extracting spatial characteristics. This combination significantly improves a strong, high-level, broadly applicable detection system that is less susceptible to bias and overfitting. Figure 2 shows the architecture's input layer, convolutional and pooling layers, GRU layer, dense layer, dropout layer, and final output layer.

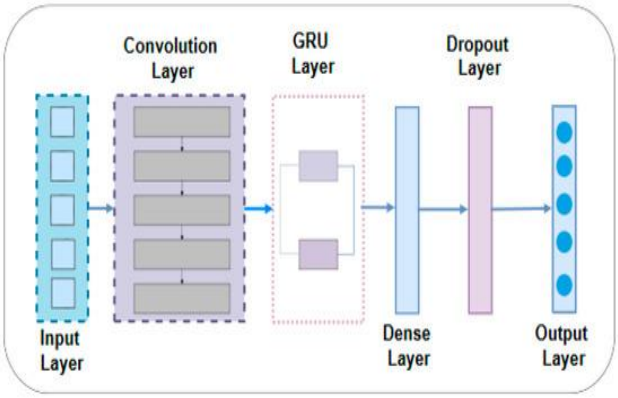

Fig 2: Architecture of the Proposed CNN-GRU Model

The input layer, which encodes the sequential information to be analyzed later, received the former. A one-dimensional convolutional layer then transforms this data using filters that identify it as a malicious object. After convolution, the results are passed through a ReLU activation unit to introduce non-linearity in the model, thereby encouraging faster learning and better gradient flow. The operation of convolution has been defined as in Equation (2):

$$y[n] = (x * w)[n] = \sum_{k=0}^{p-1} x[n+K].w[k] \qquad (2)$$

where P is the filter or kernel size, $y[n]$ is the filter or kernel value at position n, $x[n]$ is the input signal at position n, and $w[k]$ is the filter or kernel value at position s. The spatial dimensionality of the feature maps is then reduced via max pooling. By retaining the most important characteristics of each region, the process is used to increase translational invariance, reduce computational cost, and prevent overfitting. The summary features include a GRU layer. The GRU layer is designed to identify evolving attack patterns by capturing the long-range relationships and temporal dynamics of sequence data. To address the vanishing gradient problem in standard RNNs, GRUs use gating mechanisms. Two characteristics of GRU cells that control data flow are reset and update gates. Equations (3) through (6) show how the GRU is expressed mathematically.

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \qquad (3)$$
$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \qquad (4)$$
$$\tilde{h}_t = tanh(W.[r_t \times h_{t-1}, x_t]) \qquad (5)$$
$$h_t = (1 - z_t) \times h_{t-1}, z_t \times \tilde{h}_t \qquad (6)$$

The following is one approach to express these equations: x_tAs the input, $h_t$is the output vector, $\tilde{h}_t$is the activation vector, z_t is the update gate, r_tAs the reset gate, W represents matrices of weights, and the sigmoid function is represented by σ. A fully linked (dense) layer accepts the generated sequential characteristics and combines them with high-level information. A dropout layer can also be included to further reduce overfitting, randomly deleting a fraction of neurons during training to increase generalization. To arrive at the final classification decision, the output layer applies a SoftMax activation function to translate the converted attributes into a probability distribution over the two output classes, benign and malignant.

3.6. Performance Matrix

Used a variety of metrics to anticipate resource utilization. The MSE is a forecasting error statistic that uses squaring and averaging to calculate the difference between expected and actual values. As a result, the size of the prediction error affects the MSE. The RMSE is defined as the square root of the mean squared error. It is therefore less sensitive to the magnitude of errors since it takes the square root of the MSE. This allows for a more intuitive interpretation of the mistake size. The mean absolute error, or MAE, is the average of absolute errors. On the other hand, $R2$ is a statistical metric that indicates how much of the variance in the dependent variable is explained by the independent variables in a model. Its range is 0 to 1, where 0 represents no variation that the model can account for and 1 represents a perfect fit. The following are Equations (7) through (10):

$$R^2 = \frac{\sum_{i=1}^{n}(y_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \qquad (7)$$
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{\dot{P}})^2} \qquad (8)$$
$$MAE = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} |(y_i - y_i^{p})| \qquad (9)$$
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \qquad (10)$$

Where n is the number of data, $\bar{y}_i$ is the model's projected value, and yi is the actual value of real

4. Results and Discussion

The testing was conducted on a server equipped with a 3.10 GHz Xeon CPU, 16 GB of RAM, and 2 TB of storage. The algorithms were implemented using Google Colab and Python. The factors used in work are displayed in Table II. The CNN-GRU hybrid model has the highest R^2 value of 98.23%, indicating an excellent fit and predictive value, and the best accuracy with the lowest error measures (MSE: 0.0002, MAE: 0.0136, RMSE: 0.0164). GRU also performs well but does not surpass CNN on any parameter, whereas the CNN-GRU consistently stays ahead, demonstrating its efficiency in capturing complex temporal information about CPU utilization.

Table 2: Propose Models' Performance for Resource Utilisation

Matrix	CNN	GRU	CNN-GRU
MSE	0.0008	0.0005	0.0002
MAE	0.0240	0.0164	0.0136
RMSE	0.0307	0.0215	0.0164
R2 score	94.59	97.45	98.23

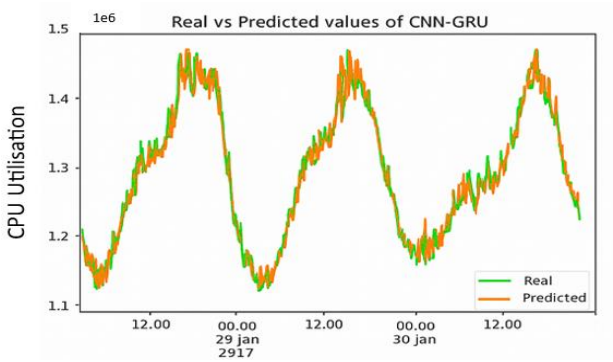

Fig 3: Line Plot For Real and Predicted Values Of CNN-GRU Model

Figure 3 illustrates a line plot of the actual and forecast CPU utilization values generated by the CNN-GRU model over two days, January 29-30. The x-axis indicates time intervals, and the y-axis depicts CPU usage between 1.1 and 1.5. The blue line shows the observed values, and the orange line shows the model forecasts. The alignment of the two curves shows that the CNN-GRU model is well-suited to learning temporal changes and the variability in CPU usage and achieves good predictive accuracy for time series.

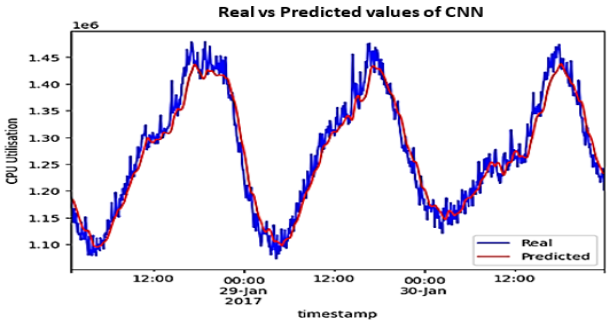

Fig 4: Line Plot For Real And Predicted Values Of CNN Model

The actual and anticipated CPU utilization numbers provided by the CNN model during the period of January 28–30, 2017, are displayed in Figure 4. The timestamp is displayed on the x-axis, and CPU utilization, which ranges from roughly 1.2 to 1.45, is displayed on the y-axis. The model's forecast is given by the red line in the plot, while the blue line shows the actual observed values. The dynamic characteristics of the two lines are similar: they are periodically shaped, with high and low points representing the CPU's usability. The fact that the predicted and actual parameters are very close indicates the effectiveness of the CNN model in capturing the short-term dynamics and trends in CPU utilization.

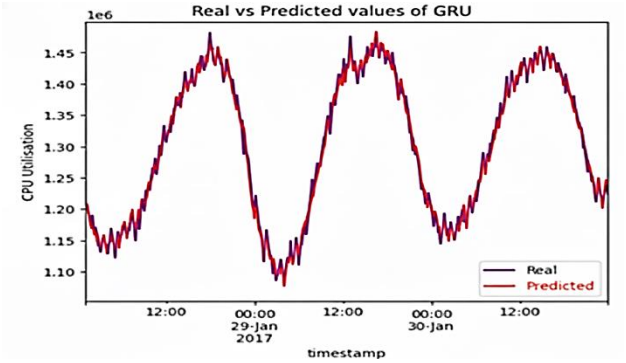

Fig 5: Line Plot for Real And Predicted Values of GRU Model

The GRU model's estimated and actual CPU utilization values over a continuous time window are compared in a line plot (Figure 5). It is evident from the plot that the predicted curve closely follows the real data, with respect to both the periodic fluctuations and the peak-to-trough trends in CPU usage. The minimal deviation between the two lines indicates that the GRU model is very effective at capturing temporal dependencies in the data and highly predictive over short time periods.

Table III presents a comparative analysis of several resource utilisation forecasting methods based on MAE and RMSE values, illustrating the predictive power of each model. Conventional methods like CP-SAE and LSTM have a relatively higher level of RMSE, which implies that they are less accurate than models based on DL. CNN and GRU perform better, with significantly lower error rates. The hybrid CNN-GRU model is the best in this selection because it minimizes MAE and RMSE, demonstrating its excellent ability to identify temporal and spatial correlations in the data. As a result, it is the best forecasting model to estimate resource use in this comparison.

Table 3: Comparison Between Different Methods for Resource Utilisation Forecasting

Models	MSE	RMSE
CP-SAE[19]	0.303	9.17
LSTM[20]	0.00045	0.0212
CNN	0.0008	0.0307
GRU	0.0005	0.0215
CNN-GRU	0.0002	0.0164

The hybrid CNN-GRU model is suggested because it successfully combines the strong temporal dependence learning strength of GRUs with the spatial feature extraction capabilities of CNNs, making it better for simulating sequential and fluctuating CPU utilization patterns. Individual models, such as CNN and GRU, can yield good performance on their own, yet CNN-GRU consistently yields lower error (MSE, MAE, RMSE) and the highest R 2, indicating its stronger capability to represent local trends and global temporal dynamics. The CNN-GRU architecture offers more consistent and precise predictions than traditional algorithms like CP-SAE and LSTM, which cannot capture temporal depth or generalise to intricate variations. The combination of this strategy guarantees minimal deviation from the actual values, accelerated convergence, and a stronger forecasting structure, which is why this method is the most accurate model in the domain of resource utilisation prediction.

5. Conclusion and Future Scope

Cloud computing enables a customer's cloud apps and services to access cloud resources dynamically and as needed. Cloud providers face challenges with future cloud resource demand, such as CPU consumption in their cloud applications to meet client requirements, as this depends on the workloads they receive. This experiment shows that the suggested CNN-GRU hybrid model achieves better results in cloud resource utilisation forecasting by effectively preserving both spatial characteristics in CNN layers and time-related dependencies in GRU units. The CNN-GRU model has the lowest MSE (0.0002), but the highest R 2 (98.23), CNN (MSE: 0.0008, R 2: 94.59) and GRU (MSE: 0.0005, R 2: 97.45). The CNN-GRU model has lower forecasting errors than the current methods (CP-SAE: RMSE: 9.17; LSTM: MSE: 0.00045), indicating high generalization and stability in learning diverse CPU utilisation patterns. The study, however, is constrained by the use of a single Microsoft Azure trace dataset and constant-time-interval inputs, which might not be representative of a heterogeneous cloud environment. Additionally, external factors such as network anomalies or workload unpredictability were not taken into account. By combining multi-cloud datasets, using dynamic time-window selection, and investigating cutting-edge architectures such as attention-based networks and transformer models, future research can overcome these restrictions and improve forecasting accuracy and scalability for real-time cloud resource management.

References

[1] G. Kaur, A. Bala, and I. Chana, "An intelligent regressive ensemble approach for predicting resource usage in cloud computing," *J. Parallel Distrib. Comput.*, 2019, doi: 10.1016/j.jpdc.2018.08.008.

[2] A. A. 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.

[3] A. Kushwaha, P. Pathak, and S. Gupta, "Review of Optimize Load Balancing Algorithms in Cloud.," *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, p. 1, 2016.

[4] T. Nguyen, N. Tran, B. M. Nguyen, and G. Nguyen, "A Resource Usage Prediction System Using Functional-Link and Genetic Algorithm Neural Network for Multivariate Cloud Metrics," in *2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA)*, IEEE, Nov. 2018, pp. 49–56. doi: 10.1109/SOCA.2018.00014.

[5] C. Liu, C. Liu, Y. Shang, S. Chen, B. Cheng, and J. Chen, "An adaptive prediction approach based on workload pattern discrimination in the cloud," *J. Netw. Comput. Appl.*, 2017, doi: 10.1016/j.jnca.2016.12.017.

[6] N. J. Kansal and I. Chana, "Artificial bee colony-based energy-aware resource utilization technique for cloud computing," *Concurr. Comput. Pract. Exp.*, 2015, doi: 10.1002/cpe.3295.

[7] S. K. Sood and R. Sandhu, "Matrix-based proactive resource provisioning in mobile cloud environment," *Simul. Model. Pract. Theory*, vol. 50, pp. 83–95, Jan. 2014, doi: 10.1016/j.simpat.2014.06.004.

[8] R. Moreno-Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, "Efficient resource provisioning for elastic Cloud services based on machine learning techniques," *J. Cloud Comput.*, 2019, doi: 10.1186/s13677-019-0128-9.

[9] 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, pp. 263–273, 2019.

[10] I. Gupta, M. S. Kumar, and P. K. Jana, "Efficient Workflow Scheduling Algorithm for Cloud Computing System: A Dynamic Priority-Based Approach," *Arab. J. Sci. Eng.*, 2018, doi: 10.1007/s13369-018-3261-8.

[11] 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 & IoT (HONET-ICT)*, IEEE, Oct. 2018, pp. 38–42. doi: 10.1109/HONET.2018.8551339.

[12] S. Gokulraj and B. G. Geetha, "Integration of firefly optimization and Pearson service correlation for efficient cloud resource utilization," *Int. J. Commun. Syst.*, vol. 31, no. 15, Oct. 2018, doi: 10.1002/dac . 3771.

[13] J. Kumar and A. K. Singh, "Cloud Resource Demand Prediction using Differential Evolution based Learning," in *2019 7th International Conference on Smart Computing and Communications, ICSCC 2019*, 2019. doi: 10.1109/ICSCC.2019.8843680.

[14] Y. Wang, Y. Wen, Y. Zhang, and J. Chen, "An Extensible Toolkit for Resource Usage Prediction in Clouds," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, Jul. 2019, pp. 466–470. doi: 10.1109/iThings/GreenCom/CPSCom/SmartData . 2019.00096.

[15] Z. Li, X. Ma, J. Li, Q. Qiu, and Y. Wang, "Efficient cloud resource management using neuromorphic modeling and prediction for virtual machine resource utilization," in *2019 IEEE International Conference on Embedded Software and Systems, ICESS 2019*, 2019. doi: 10.1109/ICESS.2019.8782503.

[16] R. Shaw, E. Howley, and E. Barrett, "A predictive anti-correlated virtual machine placement algorithm for green cloud computing," in *Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2018*, 2018. doi: 10.1109/UCC.2018.00035.

[17] M. Hassan, H. Chen, and Y. Liu, "DEARS: A Deep Learning Based Elastic and Automatic Resource Scheduling Framework for Cloud Applications," in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, IEEE, Dec. 2018, pp. 541–548. doi: 10.1109/BDCloud . 2018.00086.

[18] H. Lyu, P. Li, R. Yan, A. Masood, B. Sheng, and Y. Luo, "Load forecast of resource scheduler in cloud architecture," in *2016 International Conference on Progress in Informatics and Computing (PIC)*, IEEE, Dec. 2016, pp. 508–512. doi: 10.1109/PIC.2016.7949553.

[19] Q. Zhang, L. T. Yang, Z. Yan, Z. Chen, and P. Li, "An Efficient Deep Learning Model to Predict Cloud Workload for Industry Informatics," *IEEE Trans. Ind. Informatics*, vol. 14, no. 7, pp. 3170–3178, Jul. 2018, doi: 10.1109/TII.2018.2808910.

[20] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, "Host load prediction with long short-term memory in cloud computing," *J. Supercomput.*, vol. 74, no. 12, pp. 6554–6568, Dec. 2018, doi: 10.1007/s11227-017-2044-4.

[21] Polu, A. R., Buddula, D. V. K. R., Narra, B., Gupta, A., Vattikonda, N., & Patchipulusu, H. (2021). Evolution of AI in Software Development and Cybersecurity: Unifying Automation, Innovation, and Protection in the Digital Age. Available at SSRN 5266517.

[22] Padur, S. K. R. (2020). From centralized control to democratized insights: Migrating enterprise reporting from IBM Cognos to Microsoft Power BI. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol, 6(1), 218-225.

[23] Bitkuri, V., Kendyala, R., Kurma, J., Mamidala, V., Enokkaren, S. J., & Attipalli, A. (2021). Systematic Review of Artificial Intelligence Techniques for Enhancing Financial Reporting and Regulatory Compliance. International Journal of Emerging Trends in Computer Science and Information Technology, 2(4), 73-80.

[24] Padur, S. K. R. (2019). Machine learning for predictive capacity planning: Evolution from analytical modeling to autonomous infrastructure. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(5), 285-293.

[25] Attipalli, A., Enokkaren, S., BITKURI, V., Kendyala, R., KURMA, J., & Mamidala, J. V. (2021). Enhancing Cloud Infrastructure Security Through AI-Powered Big Data Anomaly Detection. Available at SSRN 5741305.

[26] Singh, A. A. S., Tamilmani, V., Maniar, V., Kothamaram, R. R., Rajendran, D., & Namburi, V. D. (2021). Predictive Modeling for Classification of SMS Spam Using NLP and ML Techniques. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 2(4), 60-69.

[27] Padur, S. K. R. (2020). AI augmented disaster recovery simulations: From chaos engineering to autonomous resilience orchestration. International Journal of Scientific Research in Science, Engineering and Technology, 7(6), 367-378.

[28] Reddy Padur, S. K. (2021). From Scripts to Platforms-as-Code: The Role of Terraform and Ansible in Declarative Infrastructure Rollouts. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 621-628.

[29] Kothamaram, R. R., Rajendran, D., Namburi, V. D., Singh, A. A. S., Tamilmani, V., & Maniar, V. (2021). A Survey of Adoption Challenges and Barriers in Implementing Digital Payroll Management Systems in Across Organizations. International Journal of Emerging Research in Engineering and Technology, 2(2), 64-72.

[30] Padur, S. K. R. (2018). Autonomous cloud economics: AI driven right sizing and cost optimization in hybrid infrastructures. International Journal of Scientific Research in Science and Technology, 4(5), 2090-2097.

[31] Rajendran, D., Namburi, V. D., Singh, A. A. S., Tamilmani, V., Maniar, V., & Kothamaram, R. R. (2021). Anomaly Identification in IoT-Networks Using Artificial Intelligence-Based Data-Driven Techniques in Cloud Environmen. International Journal of Emerging Trends in Computer Science and Information Technology, 2(2), 83-91.

[32] Padur, S. K. R. (2021). Bridging Human, System, and Cloud Integration through RESTful Automation and Governance. the International Journal of Science, Engineering and Technology, 9(6).

[33] Attipalli, A., BITKURI, V., KURMA, J., Enokkaren, S., Kendyala, R., & Mamidala, J. V. (2021). A Survey of Artificial Intelligence Methods in Liquidity Risk Management: Challenges and Future Directions. Available at SSRN 5741342.

[34] Padur, S. K. R. (2021). From Control to Code: Governance Models for Multi-Cloud ERP Modernization. International Journal of Scientific Research & Engineering Trends, 7(3).

[35] Routhu, K. K. (2021). Harnessing AI Dashboards in Oracle Cloud HCM: Advancing Predictive Workforce Intelligence and Managerial Agility. International Journal of Scientific Research & Engineering Trends, 7(6).

[36] Padur, S. K. R. (2021). Deep learning and process mining for ERP anomaly detection: Toward predictive and self-monitoring enterprise platforms. Available at SSRN 5605531.