Volume: 13 Issue: III Month of publication: March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67534

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Ensemble-Based Cloud Workload Prediction Using Recent AI and ML Methods for Optimized Resource Management & Scheduling

Kirtikumar J. Sharma[1], Dr. Narendra M. Patel[2]
[1]*Research Scholar, Gujarat Technological University, Gujarat, India*
[2]*Professor, Computer Engineering Department, BVM Engineering College, V. V. Nagar-388120, India*

Abstract: with the rising demand for efficient cloud computing and resource management, precise workload prediction has become vital. This paper explores altered methods used for workload predicting, from traditional methods to recent machine learning methods. We train models such as XGBoost, LightGBM, CatBoost, LSTM, and GRU, along with an ensemble method, to know their efficiency in practical cloud environments. The study uses the Alibaba Cluster 2017 dataset, focusing on batch (offline) workloads for well prediction precision. Numerous pre-processing methods, with outlier detection, normalization, and sequence creation, are applied to increase model performance. We associate the results of distinct models and ensemble methods using performance parameters like Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The results show that whereas deep learning models seizure sequential patterns, ensemble techniques deliver improved complete stability and correctness. This research shows the importance of merging multiple models to improve workload predicting and increase cloud resource consumption.

Keywords: Cloud Workload Prediction, Ensemble Learning, XGBoost, LightGBM, CatBoost, LSTM, GRU, Machine Learning, Deep Learning, Resource Management, Scheduling

I. INTRODUCTION

Workload prediction in cloud computing environments is vital for effective resource management, reducing latency, and enhancing energy consumption. As workloads become more complex, modern estimation approaches struggle to uphold precision. This study defines use of recent AI/ML methods for workload prediction, aiming on their practical applications in cloud setups[1], [2]. We work on both individual models (XGBoost, LightGBM, CatBoost, LSTM, and GRU) and an ensemble method that mixes these methods to increase forecast accuracy[3], [4], [5].

II. LITERATURE SURVEY ON WORKLOAD PREDICTION

Modern approaches, such as statistical modelling (ARIMA and time-series forecasting) and rule-based heuristics, have been extensively used for workload forecast. Though, these methods frequently struggle in dynamic situations with highly changing workloads[6].

A. AI and ML-based Workload Prediction Methods

Supervised learning models, like Random Forest, XGBoost, CatBoost, LightGBM, and deep learning-based neural networks, have presented ability in predicting workload configurations with upper accuracy. Another group is unsupervised learning models, where clustering methods like K-Means and DBSCAN assistance to identify workload trends lacking of labelled records. Recurrent Neural Networks (RNNs) and Gated Recurrent Units (GRUs) are effective in treatment sequential workload information by taking long-term dependencies[7]. Reinforcement learning-based methods, like adaptive workload management using Q-learning and deep reinforcement learning (Meta-RHDC), improve scheduling and dynamic load harmonizing. Hybrid models, which association machine learning techniques through optimization algorithms (e.g., Lyrebird Falcon Optimization), increase resource utilization and system performance[8], [9].

In recent years, ensemble approaches, which mix models like XGBoost, LightGBM, CatBoost, LSTM, and GRU, have improved forecast strength and precision in cloud environments. AI-driven resource allocation policies incorporate predictive models with actual scaling mechanisms, sanctioning cloud platforms to dynamically correct workloads and improve cost efficiency[10].

B. Challenges and Future Directions

AI models must evolve dynamically to adapt to real-time workload fluctuations. Furthermore, dataset preprocessing—such as handling missing values and standardizing features—remains a critical challenge. For better resource utilization and energy efficiency, carbon-aware scheduling techniques should be integrated to support sustainable cloud computing.

In today's cloud landscape, the need for AI-driven workload prediction has grown, requiring scalability across heterogeneous cloud environments. Additionally, benchmarking individual models (XGBoost, LightGBM, CatBoost, LSTM, and GRU) against their ensemble counterparts is essential for evaluating performance improvements.AI and ML approaches for workload prediction in cloud environments

Deep learning models play a crucial role in workload prediction. Long Short-Term Memory (LSTM) networks are nominal in capturing time-based dependencies, while Gated Recurrent Units (GRUs), a modified of LSTM, offer better computational efficiency for workload predicting. Convolutional Neural Networks (CNNs) are beneficial for identifying workload patterns in multidimensional data. Transformer models are developing as powerful tools for actual workload forecasting, providing improved scalability and forecasted accuracy.

Gradient boosting algorithms are widely used for workload prediction. XGBoost is a highly efficient gradient boosting method, particularly suited for structured workload prediction. LightGBM is enhanced for speed and competence, making it actual for handling extensive workload data. CatBoost bests in managing unconditional features, dropping the need for broad physical pre-processing. Reinforcement Learning methods can be use similar to Q-Learning-Based Scheduling: Increases decision-making for vibrant cloud workload scaling and Reinforcement Learning for Dynamic Load Harmonizing: Enhances system consistency and cuts congestion in cloud surroundings.

Hybrid AI methods can be used to improve workload forecast and optimization. Mixing neural networks with optimization algorithms, like Lyrebird Falcon Optimization, recovers predictive accuracy. AI-enhanced auto-scaling policies help reduce cost ineffectiveness while refining overall cloud workload performance. Moreover, an ensemble model that combines XGBoost, LightGBM, CatBoost, LSTM, and GRU improves workload prediction correctness and adaptability compared to individual models.

III. METHODOLOGY

In this paper, we use three models for workload predicting. The first approach includes powerful machine learning models, with LightGBM, XGBoost, and CatBoost. The second emphases on deep learning models, exactly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Lastly, we employ an ensemble-based analytical modeling method that mixes multiple traditional machine learning models (LightGBM, XGBoost, and CatBoost) to improve prediction accuracy.

A. Machine Learning Models

We assess and visualize the performance of three dominant machine learning models—LightGBM, XGBoost, and CatBoost—for forecasting num_instances (the number of instances in the cloud dataset). These models are precisely used for regression tasks. The dataset is divided into 80% training and 20% testing sets. We use random_state=42 to ensure reproducibility. The machine learning models are then modified and trained as follows:

Each model is trained on X_train and y_train, and forecasts are made on y_test using the trained models. Error metrics, with Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), are calculated using Equation 1 and Equation 2. Lesser RMSE and MAE values specify improved model performance.

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

B. Deep Learning Models

We use the application of deep learning models, precisely Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), for time-series workload forecast. The results show that both LSTM and GRU successfully capture sequential patterns in cluster workload instances. Though, GRU somewhat disappoints compared to LSTM, showing higher Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE)*[5]*. Further adjustment of hyper parameters or employing ensemble methods could improve predictive performance.

- Features (X): All columns except num_instances.
- Target (y): The num_instances column (dependent variable).
- Normalization: MinMaxScaler scales all numerical values between [0,1]
- Splits dataset into: 80% Training (X_train, y_train).20% Testing (X_test, y_test).

1) LSTM Model Architecture

- LSTM layer: 64 neurons, ReLU activation.
- Dense layer: 32 neurons, ReLU activation.
- Output layer: 1 neuron (predicts num_instances).

2) Training Parameters

- Epochs: 20
- Batch size: 64
- Validation split: 10% of training data used for validation.
- Optimizer: Adam (efficient weight optimization).
- Loss function: Mean Squared Error (MSE).

C. Ensemble Approaches

This paper evaluates an ensemble-based predictive modeling approach for workload forecasting. The ensemble integrates multiple traditional machine learning models (LightGBM, XGBoost, and CatBoost) with deep learning models (LSTM and GRU) to leverage their complementary strengths. The proposed approach enhances predictive accuracy by reducing model-specific biases and improving generalization. Performance metrics, such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), indicate that the ensemble method outperforms individual models in terms of stability and accuracy.

1) Deep Learning Models: LSTM, GRU (from TensorFlow/Keras).

LSTM Model Architecture

- Layer: 64 neurons, relu activation.
- Output Layer: Single neuron for prediction.
- Training Parameters:
- Epochs: 20
- Batch size: 64
- Loss function: MSE

GRU Model Architecture

- activation=relu
- optimizer=adam
- epochs=20
- batch_size=64

2) Machine Learning Models: LightGBM, XGBoost, CatBoost (gradient boosting models).

- lgb_model = lgb.LGBMRegressor(random_state=42).fit(X_train, y_train)
- xgb_model = xgb.XGBRegressor(random_state=42).fit(X_train, y_train)
- cat_model = CatBoostRegressor(random_state=42, verbose=False).fit(X_train, y_train)

Train-Test Split: 80% data used for training, 20% for testing.

3) Combines all models' predictions using a simple weighted average

- ensemble_pred = (lstm_pred.flatten() + gru_pred.flatten() + lgb_pred + xgb_pred + cat_pred) / 5 ---(3)

IV. IMPLEMENTATION AND RESULTS

A. Dataset

The Alibaba Cluster 2017 [6] trace dataset provides a comprehensive view of real-world cloud workloads, making it a valuable resource for cloud computing research. Collected over an 8-day period from a production cluster, it represents the operational data of more than 4,000 machines with different configurations[11]. This large-scale cloud cluster trace is particularly useful for investigating cloud resource management, scheduling algorithms, and cluster optimization. The dataset contains two major types of workloads: offline and online workloads. All online jobs arrive at the start time, so for prediction, we focus on the offline workload file (batch workload). As shown in Table 1, the batch_task file contains eight columns and includes 80,554 requests. The task creation time and end time are used to calculate task duration, while key features such as instances, CPU requested, memory requested, and job ID are considered for analysis.

TABLE IBATCH_TASK.CSV

task_create_time	task_end_time	job_id	task_id	no_of_instance	status	req_cpu	req_memory
21669	21695	44	249	1	Terminated	50	0.004074
24775	24927	61	441	93	Running	50	0.007977636
6036	6046	4	7	393	Waiting		
…	….	…	…	…	…	…	…

B. Pre-processing

For outlier detection, it is essential to ensure that extreme values do not distort predictions. The following methods are used for this purpose: Boxplots, Z-score, and the Interquartile Range (IQR) method. For normalization and scaling, the selected columns are standardized to maintain a mean of 0 and a standard deviation of 1. If additional data is added later, the same scaling must be applied to maintain consistency. For sequence creation, which is required for sequential models such as LSTM and GRU, the data is converted into a sequential format by creating sequences or using a sliding window approach. For the train-test split, the dataset is clearly divided into training and testing sets, typically following an 80%-20% ratio, ensuring a proper evaluation of model performance.

C. Result

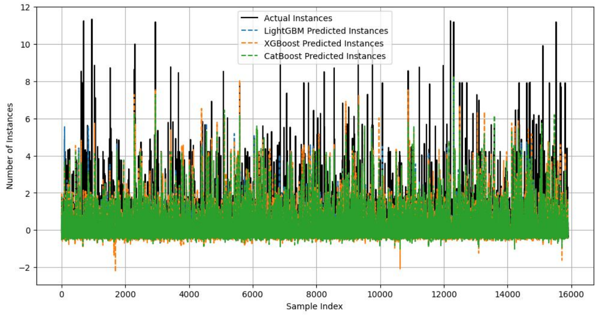

Fig. 1 Actual Vs Predicted Instances LightGBM, XGBosst, CatBoost Model

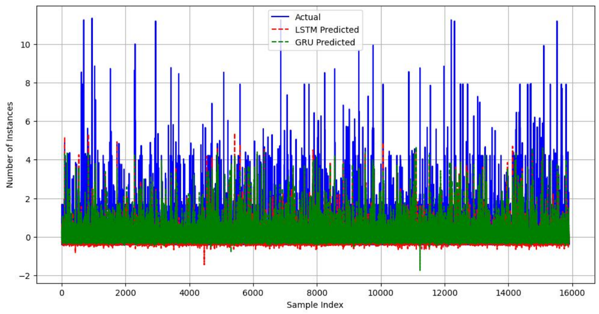

Fig.2 Actual Vs Predicted Instances LSTM and GRU Model

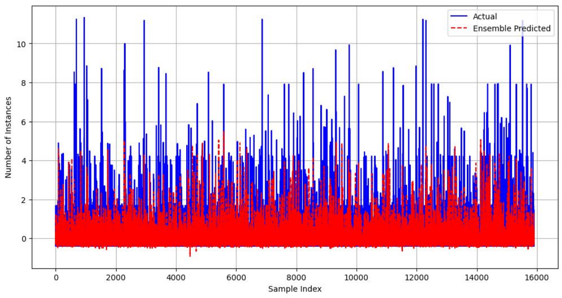

Fig. 3 Actual Vs Predicted Instances LSTM and GRU Model

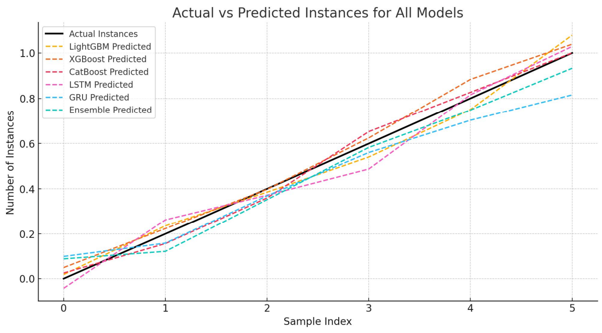

Fig. 4 Actual Vs Predicted Instances for All Models

Figure 1 illustrates the output of Actual vs. Predicted Instances for LightGBM, XGBoost, and CatBoost. Figure 2 presents the Actual vs. Predicted Instances for the LSTM and GRU deep learning models. Figure 3 depicts the Actual vs. Predicted Instances for the ensemble method, highlighting its performance compared to individual models. Figure 4 provides a comprehensive visualization of Actual vs. Predicted Instances across all models, including LightGBM, XGBoost, CatBoost, LSTM, GRU, and the Ensemble model. This assessment helps assess how carefully each model's forecasts bring into line with the real values.

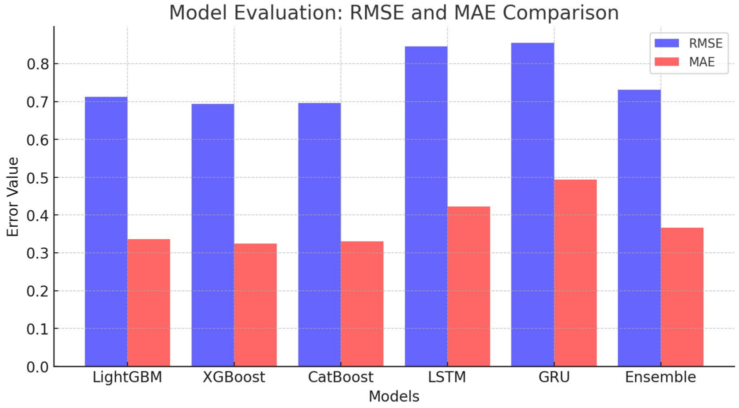

Fig. 5 Model Evaluation RMSE and MAE Comparison

Table II offerings a numerical summary of the performance of all models, providing key assessment metrics. Figure 5 visualizes this assessment through a bar chart, showing the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) through diverse models. This graphical image assistances in measuring the comparative performance and accuracy of each model.

TABLE III
PERFORMANCE SUMMARY (LOWER RMSE & MAE IS BETTER)

Model	RMSE	MAE	Performance
LightGBM	0.7128	0.3359	✅Good
XGBoost	0.6942	0.3241	⭐ Best
CatBoost	0.6961	0.3296	✅Good
LSTM	0.8451	0.4217	❌High Error
GRU	0.8548	0.4939	❌High Error
Ensemble	0.7309	0.3662	✅Better than LSTM/GRU

V. CONCLUSIONS

This study discovers the development in workload prediction using AI and machine learning methods, importance their real time applications in cloud surroundings. Compared to old-style methods, deep learning, reinforcement learning, and hybrid AI models demonstrate substantial growths in accuracy and efficiency. The use of progressive models like GRU, CatBoost, LightGBM, and XGBoost—together separately and within an ensemble—reveals the potential for improved workload predicting and cloud resource management.

Future research should emphasis on additional benchmarking these models, improving their compliance to real-time workload variations, and mixing energy-efficient policies. Moreover, emerging scalable AI-driven workload prediction structures will be crucial for enhancing modern cloud infrastructures and improving complete cloud system performance.

VI. ACKNOWLEDGMENT

We sincerely acknowledge the All India Council for Technical Education (AICTE) for funding the RPS project no 8-43 /FDC /RPS (POLICY-1) / 2019-20 titled "A Vision Based Dynamic Gesture Recognition of Indian Sign Language using Kinect." The infrastructure resources, including the server used for this execution, were acquired under this scheme.

REFERENCES

[1] M. Cate, “AI and Machine Learning for Dynamic Workload Orchestration”.

[2] L. Harris, “AI-Based Workload Prediction for Energy-Efficient Cloud Resource Allocation,” ResearchGate, 2024.

[3] S. Reports, “A Hybrid Cloud Load Balancing and Host Utilization Prediction Method Using Deep Learning with Particle Swarm Intelligence and Genetic Algorithm,” Sci. Rep., 2023.

[4] I. R. E. Journals, “AI-Powered Predictive Scaling in Cloud Computing,” IRE J., 2023.

[5] A. Setayesh, H. Hadian, and R. Prodan, “An Efficient Online Prediction of Host Workloads Using Pruned GRU Neural Nets,” ArXiv Prepr. ArXiv230316601, 2023.

[6] “Alibaba cluster-trace-v2017.” [Online]. Available: https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2017

[7] D. Saxena, J. Kumar, A. K. Singh, and S. Schmid, “Performance Analysis of Machine Learning Centered Workload Prediction Models for Cloud,” ArXiv Prepr. ArXiv230202452, 2023.

[8] G. Krishnan, “ENHANCING CLOUD COMPUTING PERFORMANCE THROUGH AI-DRIVEN DYNAMIC RESOURCE ALLOCATION AND AUTO-SCALING STRATEGIES”.

[9] M. Xu, C. Song, H. Wu, S. S. Gill, K. Ye, and C. Xu, “EsDNN: Deep Neural Network based Multivariate Workload Prediction Approach in Cloud Environment,” ArXiv Prepr. ArXiv220302684, 2022.

[10] A. Rossi, A. Visentin, D. Carraro, S. Prestwich, and K. N. Brown, “Forecasting Workload in Cloud Computing: Towards Uncertainty-Aware Predictions and Transfer Learning,” ArXiv Prepr. ArXiv230313525, 2023.

[11] K. J. Sharma and N. M. Patel, “Investigation of Alibaba Cloud Data Set for Resource Management in Cloud Computing,” 2021, doi: 10.1007/978-3-030-76776-1_15.