JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Log-Transformed Regime-Based Prediction of Cloud Job Length Using Machine Learning

Ardi Pujiyanta [1,*], Bambang Robiin [2], and Faisal Fajri Rahani [2]

[1] Master of Electrical Engineering Study Program, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta 55166, Indonesia; e-mail : ardipujiyanta@tif.uad.ac.id
[2] Department of Informatics, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta 55166, Indonesia; e-mail : bambang.robiin@tif.uad.ac.id; faisal.fajri@tif.uad.ac.id
* Corresponding Author : Ardi Pujiyanta

Abstract: Cloud job-length prediction remains challenging when the target distribution is highly skewed and contains rare extreme values. This study proposes a log-transformed, regime-based machine learning framework for robust prediction of cloud job length, represented in million instructions (MI). The approach integrates sequential feature engineering, logarithmic target transformation, weighted learning, and regime-aware modeling to distinguish between normal and extreme job-length behavior. Using an ordered GoCJ-derived cloud job-length sequence of 1000 jobs, the dataset exhibits a heavy-tailed distribution, with a mean of 129,662 MI, a median of 93,000 MI, a 95th percentile of 525,000 MI, a 99th percentile of 900,000 MI, and a skewness of 3.695. The proposed model is evaluated against sequential baselines and stronger machine learning baselines, including Naive_Last, RollingMean_5, Global_Log_ExtraTrees, RandomForest, GradientBoosting, and MLP_Log. On the main test split, the proposed Regime_Log_ExtraTrees achieved the best RMSE of 206,255.66 and the least negative R^2 of −0.01062, while Global_Log_ExtraTrees remained competitive in terms of MAE, MedAE, and RMSLE. Additional walk-forward validation confirms that the regime-aware model consistently achieves the best mean RMSE and mean R^2 across temporal folds. Ablation results further show that regime-aware learning is the primary contributor to robustness, although accurate prediction of extreme jobs remains challenging. These findings indicate that log-transformed, regime-based learning provides a practical and more robust strategy for cloud job-length prediction under heavy-tailed workload conditions.

Keywords: Cloud computing; Cloud job-length prediction; Energy-efficient computing; Heavy-tailed data; Log transformation; Machine learning; Regime-based learning; Workload prediction.

Received: March, 15th 2026
Revised: April, 16th 2026
Accepted: April, 19th 2026
Published: April, 21st 2026

1. Introduction

Cloud computing has become a dominant platform for delivering scalable and on-demand computing services. However, efficient cloud operation still depends heavily on the ability to characterize and predict workload behavior for improved resource allocation and scheduling decisions. In this context, cloud job-length prediction has emerged as an important research problem because it directly influences scheduling quality, resource utilization, service-level agreement (SLA) compliance, and overall system responsiveness [1]–[4]. Previous studies have explored a wide range of workload prediction approaches in cloud environments, including time-series forecasting, support vector regression, CNN-LSTM, Bi-LSTM hybrids, multivariate deep neural networks, attention-based models, and adaptive hyperparameter tuning strategies [4]–[11]. These approaches demonstrate that machine learning can significantly enhance predictive decision-making in dynamic cloud environments. In addition, several studies have linked predictive modeling with scheduling and resource management, highlighting the practical value of workload-aware and prediction-based cloud optimization [9], [10], [12], [13]. In the broader cloud-optimization context, recent studies published in the same journal have also examined task scheduling with dynamic resource allocation and clustering-based optimization, as well as service-broker and load-balancing policies for improving cloud application performance. Although these studies do not explicitly model cloud job length, they reinforce the importance of intelligent decision-support mechanisms for cloud resource management and performance improvement [14], [15].

Despite these advances, several limitations remain. First, many prior studies focus on general workload forecasting, host utilization, or broader scheduling optimization rather than explicitly targeting cloud job-length prediction as a primary task [9], [16]–[19]. Second, although existing machine learning models often achieve promising average performance, their reliability tends to degrade when the target distribution is highly skewed and contains rare but influential extreme values [18], [20]–[22]. Third, completion-time and runtime-oriented studies are often conducted in HPC or integrated cloud–HPC environments, which are not fully equivalent to the cloud job-length prediction setting addressed in this work [16], [19], [23], [24]. It is therefore important to distinguish cloud job-length prediction from both general workload prediction and execution-time estimation. General workload prediction typically focuses on aggregate demand, utilization, or system-level workload intensity, whereas execution-time estimation often relies on richer application-level or runtime-specific context, particularly in HPC-oriented environments. In contrast, this study targets per-job cloud job length as a distinct predictive variable that directly affects scheduling priority, queue dynamics, and resource-allocation decisions in cloud systems, especially under heavy-tailed job distributions.

To address these challenges, this study proposes a log-transformed, regime-based machine learning framework for cloud job-length prediction. The core idea is to stabilize the target distribution through logarithmic transformation and to explicitly model different workload regimes so that normal and extreme job-length patterns can be handled more effectively. Unlike conventional global predictors that apply a single regression function across all samples, the proposed approach introduces regime-aware learning and weighted training to better capture the asymmetric behavior of heavy-tailed data. The framework is evaluated against simple sequential baselines and strong non-regime machine learning baselines to examine whether regime-aware learning can provide more robust predictive performance under highly variable cloud workload conditions [2], [3], [18], [21]. The main contributions of this paper are summarized as follows:

- Formulates cloud job-length prediction as a machine learning problem under heavy-tailed workload conditions and highlights logarithmic target transformation as a practical strategy for stabilizing prediction.
- Proposes a regime-based learning framework that explicitly separates normal and extreme job-length patterns to improve robustness under skewed target distributions.
- Evaluates the proposed method against both naive sequential baselines and non-regime machine learning models, including Random Forest, Gradient Boosting, and global ExtraTrees-based prediction, ensuring a controlled and fair comparison.
- Analyzes the trade-off between large-error reduction and average-case predictive stability using RMSE, MAE, MedAE, R^2, and RMSLE, demonstrating that the primary contribution lies in robustness rather than universal superiority across all metrics.

The rest of this paper is organized as follows. Section 2 reviews related studies on cloud workload prediction and prediction-based scheduling. Section 3 presents the proposed log-transformed regime-based prediction framework. Section 4 describes the experimental setup, dataset representation, feature engineering, and evaluation metrics. Section 5 discusses the experimental results and comparative analysis. Finally, Section 6 concludes the paper and outlines future research directions.

2. Literature Review

2.1. Cloud Workload and Resource Prediction

Cloud workload prediction has received significant attention because workload variability directly affects scheduling quality, resource utilization, and service-level agreement (SLA) compliance. Recent studies have explored time-series and machine learning approaches for workload characterization in cloud environments. Time-series forecasting models for cloud datacenter workload prediction were examined in [7], while CNN–LSTM-based architectures were adopted in [10] and [22] to capture temporal dependencies in resource utilization and virtual machine workloads. Similar deep learning approaches have also been reported in [5], [6], [8], [10], [22], [25], where deep and hybrid neural models were used to improve predictive accuracy under dynamic workload patterns.

Several studies have extended workload prediction beyond single-model learning. A multivariate attention-based approach was introduced in [9], while a deep neural framework for multivariate cloud workload prediction was proposed in [6]. More recent work has incorporated adaptive and ensemble strategies. Ensemble learning for workload prediction was explored in [12], and automated hyperparameter tuning for adaptive cloud prediction was emphasized in [11]. In addition, uncertainty-aware forecasting and transfer learning were highlighted in [2], indicating that workload prediction should not only be accurate on average but also robust under changing operating conditions.

Other related studies have focused on predictive estimation of resource behavior and cloud management variables. Support vector regression was used for host utilization prediction in [4], while predictive modeling has been linked with SLA-oriented and resource optimization objectives in [21] and [26]. Joint workload forecasting and energy-state estimation were investigated in [13], and adaptive workload management for SLA compliance and resource optimization was further discussed in [26]. These studies consistently demonstrate the usefulness of machine learning-based prediction for cloud decision-making; however, most of them focus on workload intensity, host utilization, or general resource behavior rather than the more specific problem of cloud job-length prediction.

2.2. Prediction-Based Scheduling and Runtime-Oriented Studies

Prediction has also been widely integrated into cloud scheduling research. A systematic review of prediction-based scheduling techniques for cloud workloads was presented in [3], showing that predictive information can improve scheduling effectiveness, although the surveyed methods differ in prediction target, decision scope, and operational complexity. Related scheduling-oriented studies include hybrid prediction-based scheduling [27], deep reinforcement learning for resource scheduling [28], and Deep Q-LSTM-based workload scheduling [17]. A broader review of AI-driven job scheduling in cloud computing was also provided in [29], confirming that prediction is increasingly treated as a key enabling component for intelligent scheduling.

A related research direction focuses on runtime or completion-time prediction. Runtime prediction in HPC settings was studied in [23], [30], while completion-time estimation for machine learning jobs was examined in [16]. Runtime and resource utilization prediction in integrated cloud–HPC systems was investigated in [19]. These studies are methodologically relevant because they highlight the importance of estimating job behavior prior to execution. However, their problem setting differs from that of the present study. HPC runtime prediction and predictability-centric scheduling typically rely on application-level or system-level context, which is not directly comparable to the cloud job-length prediction scenario addressed in this work.

2.3. Research Gap and Position of the Present Study

Although the literature shows substantial progress in cloud workload prediction, several gaps remain. First, most existing studies focus on general workload forecasting, host utilization prediction, SLA-oriented prediction, or scheduling enhancement, rather than directly modeling cloud job length as the primary prediction target [2]–[5], [9], [19]. Second, many approaches rely on a single global predictor, even though cloud job-length data often exhibit positively skewed and heavy-tailed behavior with occasional extreme values [18], [21], [22]. Under such conditions, improvements in average-case accuracy do not necessarily translate into reliable prediction of rare but operationally critical extreme jobs. Third, recent studies increasingly emphasize robustness, uncertainty awareness, and adaptability under dynamic cloud conditions [2], [20], [24], [26], suggesting that prediction frameworks should explicitly account for heterogeneous target regimes.

Based on these observations, the present study is positioned at the intersection of cloud workload prediction and prediction-aware resource management, with a specific focus on cloud job-length prediction under heavy-tailed behavior. Unlike prior studies that emphasize host utilization, aggregate workload demand, or HPC runtime estimation, this work models cloud job length using a log-transformed, regime-based machine learning framework. The proposed design combines logarithmic target transformation, regime-aware learning, and weighted training, and is evaluated against both sequential baselines and strong non-regime ensemble models. Accordingly, the contribution of this study lies in improving robustness under heavy-tailed conditions rather than claiming universal superiority across all prediction tasks.

Table 1. Qualitative positioning of the present study against representative related directions

Study Direction	Main Target	Typical Methods	Log-Transformed Target	Extreme-Aware Handling	Regime-Aware Learning
General cloud workload prediction [2], [6], [10], [22], [25], [26]	Workload / resource demand	CNN–LSTM, Bi-LSTM, DNN, time-series models	Limited	Limited	No
Resource utilization and SLA prediction [4], [13], [21], [26]	Utilization, SLA, host behavior	SVR, deep learning, hybrid ML	Limited	Indirect	No
Prediction-based scheduling [3], [17], [27]–[29]	Scheduling quality and resource decisions	DRL, hybrid scheduling, review-based frameworks	Limited	Indirect	No
Runtime / completion-time prediction [16], [19], [23], [30]	Runtime or completion time	Runtime modeling, tandem prediction, scheduling-centric estimation	Partial	Partial	No
Present study	Cloud job length	Log-transformed, weighted, regime-based ML	Yes	Yes	Yes

Table 1 highlights a clear structural gap in the existing literature. Most prior studies focus on workload-level or system-level prediction, with limited attention to job-level prediction under heavy-tailed distributions. Moreover, although some methods implicitly address extreme values, explicit modeling of heterogeneous regimes remains largely absent. In contrast, the present study integrates logarithmic transformation, regime separation, and weighted learning within a unified framework. This design enables a more targeted treatment of heavy-tailed behavior, particularly for rare but high-impact job-length observations, and emphasizes robustness at the job level rather than solely improving average predictive accuracy.

3. Proposed Method

This study proposes a log-transformed, regime-based machine learning framework for cloud job-length prediction. The method is designed for sequential cloud job-length data in which the target distribution is highly skewed and contains rare extreme values. Recent studies have shown that cloud prediction performance can be improved through temporal feature extraction, deep or ensemble learning, and prediction-aware resource management [1]–[4], [6], [8], [10], [18], [26]. However, many existing approaches still rely on a single global predictor, which may be less robust when normal and extreme job-length patterns coexist within the same distribution [18], [21], [22]. To address this limitation, the proposed framework integrates three key components:

- logarithmic target transformation,
- regime-aware learning, and
- weighted training for extreme samples.

The overall workflow is summarized in Algorithm 1 and illustrated in Fig. 1. First, the job sequence is sorted according to its submission order and represented as a univariate cloud job-length series. Second, a feature matrix is constructed from historical observations using lag variables, rolling statistics, exponential moving averages (EMA), local ranges, relative ratios, and local deviation indicators. Third, the target variable is transformed using a logarithmic function to reduce the influence of extreme values and stabilize the prediction space. Next, a regime threshold is derived from the training set so that each sample can be categorized into either a normal regime or an extreme regime. A classifier is then trained to estimate the probability that a sample belongs to the extreme regime, while two regressors are trained separately for normal and extreme samples. Finally, the prediction is obtained by combining both regressors in the transformed space and mapping the result back to the original job-length scale.

This design differs from conventional global prediction models because it explicitly treats extreme job-length behavior as a distinct learning regime. In addition, weighted learning is introduced to ensure that rare but operationally important large job-length observations contribute more strongly during training. As a result, the proposed framework aims to reduce large prediction errors while maintaining acceptable average-case predictive stability. It should be noted that the proposed framework is not a full latent mixture-of-experts or hidden regime-switching model. Instead, it adopts a transparent, threshold-guided regime construction derived from the training distribution, followed by a classifier-assisted combination of regime-specific regressors. This design prioritizes interpretability and reproducibility under heavy-tailed cloud job-length data while avoiding overstated novelty relative to broader mixture-based modeling approaches.

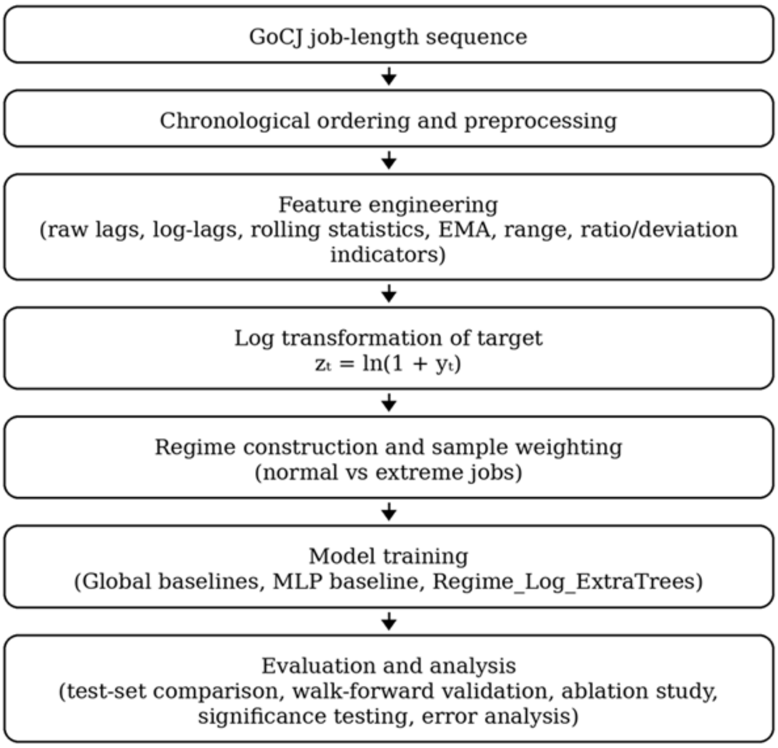


Figure 1. Workflow of the proposed log-transformed regime-based cloud job-length prediction framework.

3.1. Algorithm

The overall procedure of the proposed model is summarized in Algorithm 1.

Algorithm 1. Log-Transformed Regime-Based Prediction of Cloud Job Length

INPUT: ordered job-length sequence $Y = \{y_t\}_{t=1}^{T}$, maximum lag L, rolling windows W, regime quantile q, extreme weights λ_1 and λ_2, blending parameter α

OUTPUT: predicted cloud job length $\hat{y}_t$

1: Sort the jobs according to their sequence order.
2: Construct lag-based features from the previous Lobservations
3: Compute rolling mean, rolling standard deviation, rolling minimum, rolling maximum, rolling quantiles, exponential moving averages, local ranges, and relative ratios
4: Form the feature vector x_t using only past information to avoid future leakage.
5: Transform the target value using $z_t = \ln(1 + y_t)$.
6: Split the data into training, validation, and testing sets in time order
7: Compute the regime threshold $\tau$from the training set using quantile q.
8: Assign each training sample to the normal regime or the extreme regime based on τ.
9: Train a regime classifier to estimate the probability of extreme behavior.

Algorithm 1 *(continued)*. Log-Transformed Regime-Based Prediction of Cloud Job Length

10: Train a normal-regime regressor using normal samples in the log-transformed space.
11: Train an extreme-regime regressor using extreme samples in the log-transformed space.
12: Apply weighted learning so that extreme and very extreme samples receive higher training importance.
13: For each test sample, obtain the extreme probability p_t from the classifier.
14: Predict $\hat{z}_t^{(N)}$ from the normal regressor and $\hat{z}_t^{(E)}$ from the extreme regressor.
15: Combine both predictions using regime-aware blending.
16: Convert the final result back to the original scale using $\hat{y}_t = \exp(\hat{z}_t) - 1$.
17: Return the final prediction $\hat{y}_t$.

3.2. Mathematical Formulation

Let y_t denote the cloud job length of the t-th job, where $y_t > 0$. The ordered sequence is written as

$$Y = \{y_1, y_2, \dots, y_T\} \tag{1}$$

Because the target distribution is positively skewed, the proposed model first applies logarithmic transformation.

$$z_t = \ln(1 + y_t) \tag{2}$$

The input feature vector for time step tis constructed only from historical information. In a general form, the feature vector can be written as

$$x_t = [y_{t-1}, y_{t-2}, \dots, y_{t-L}, z_{t-1}, z_{t-2}, \dots, z_{t-L}, \mu_t^{(w)}, \sigma_t^{(w)}, m_t^{(w)}, M_t^{(w)}, e_t^{(w)}, r_t^{(w)}, \rho_t] \tag{3}$$

where L is the maximum lag, $\mu_t^{(w)}$ is the rolling mean for window w, $\sigma_t^{(w)}$ is the rolling standard deviation, $m_t^{(w)}$ and $M_t^{(w)}$ are the rolling minimum and maximum, $e_t^{(w)}$ is the exponential moving average, $r_t^{(w)}$ is the local range, and ρ_t denotes ratio- or deviation-based local indicators.

The regime threshold is derived from the training set using a quantile function:

$$\tau = Q_q(Y_{\text{train}}) \tag{4}$$

where $Q_q(\cdot)$denotes the q-th quantile. Based on this threshold, the regime label is defined as

$$g_t = \begin{cases} 1, & y_t > \tau \\ 0, & y_t \le \tau \end{cases} \tag{5}$$

where $g_t = 1$ indicates the extreme regime and $g_t = 0$ indicates the normal regime.

To emphasize large and rare job-length observations during training, the proposed method introduces weighted learning. The sample weight is defined as

$$w_t = \begin{cases} 1, & y_t \le \tau \\ \lambda_1, & \tau < y_t \le \tau_v \\ \lambda_2, & y_t > \tau_v \end{cases} \tag{6}$$

where τ_v is a very-extreme threshold, and $\lambda_2 > \lambda_1 > 1$.

Next, a classifier is trained to estimate the probability that sample x_t belongs to the extreme regime:

$$p_t = C(x_t) \tag{7}$$

where $C(\cdot)$is the regime classifier and $0 \le p_t \le 1$.

Two regressors are then trained in the transformed space. The first regressor predicts the log-transformed job length for the normal regime:

$$\hat{z}_t^{(N)} = f_N(x_t) \tag{8}$$

and the second regressor predicts the log-transformed job length for the extreme regime:

$$\hat{z}_t^{(N)} = f_N(x_t) \tag{9}$$

The final prediction in the transformed space is obtained using regime-aware blending:

$$\hat{z}_t = (1 - \tilde{p}_t)\hat{z}_t^{(N)} + \tilde{p}_t \hat{z}_t^{(E)} \tag{10}$$

where $\tilde{p}_t = p_t^{\alpha}$ and $\alpha > 0$ is a calibration parameter controlling the strength of regime influence. Finally, the prediction is returned to the original scale by inverse transformation:

$$\hat{y}_t = \exp(\hat{z}_t) - 1 \tag{11}$$

The proposed method is therefore a two-level prediction framework: a classifier estimates whether the current sample resembles extreme job-length behavior, while the regressors generate regime-specific predictions in a stabilized log space. This formulation is intended to improve robustness on heavy-tailed cloud job-length data and to reduce the limitations of a single global regression model [3], [18], [21], [22].

4. Experimental Setup

4.1. Dataset and Descriptive Statistics

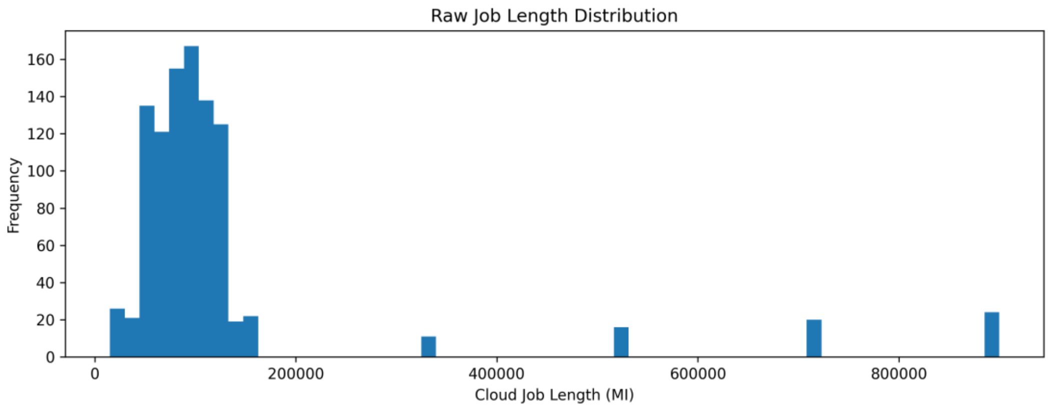
The dataset used in this study was derived from the Google Cloud Jobs (GoCJ) benchmark dataset, which was introduced as a publicly available cloud workload dataset for distributed and cloud computing research. The GoCJ dataset is archived in a public Mendeley Data repository and formally described in the Data journal article by Hussain and Aleem [31]. In this study, an ordered job-length sequence of 1000 jobs was extracted from the GoCJ source, and the target variable was represented as cloud job length in million instructions (MI). The acquisition process followed a chronological extraction of job-length records, after which the sequence was ordered according to submission order and preprocessed for sequential modeling. Each sample was constructed using only past observations to avoid future leakage. Each record therefore corresponds to a single job-length observation used for sequential prediction. In this setting, MI is used as a proxy for cloud job length rather than actual wall-clock runtime. The use of a public dataset and explicit preprocessing pipeline improves reproducibility and clarifies the experimental foundation of this study.

The target distribution is strongly right-skewed and exhibits heavy-tailed behavior. The mean job length is 129,662 MI, while the median is 93,000 MI, indicating that the upper tail substantially inflates the average. The standard deviation is 162,632.50 MI, further confirming high variability. The 95th percentile reaches 525,000 MI and the 99th percentile reaches 900,000 MI, with a skewness value of 3.695. These characteristics indicate that a small number of extremely large jobs dominate the distribution, making mean-centered modeling insufficient. This statistical property motivates the use of logarithmic transformation, regime-aware modeling, and robustness-oriented evaluation.

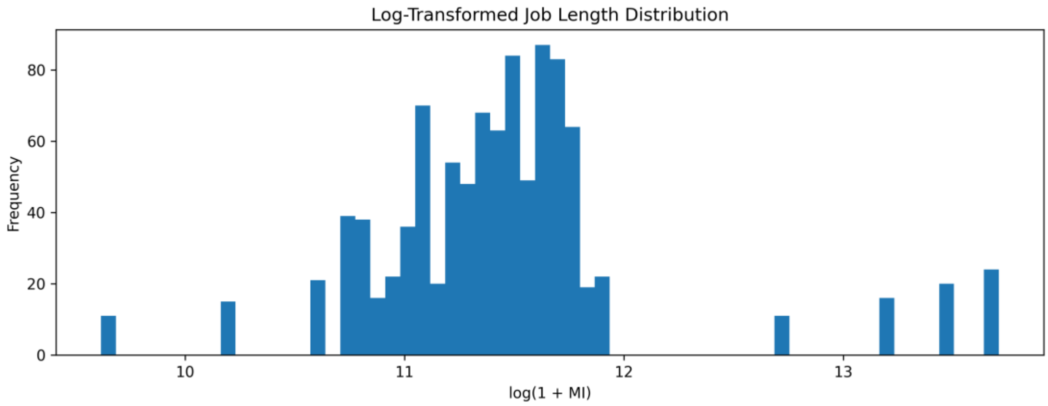
Figure 2 illustrates the target distribution before and after logarithmic transformation. The raw cloud job length is highly skewed, with most observations concentrated in the lower-to-middle range and a small number of extreme values forming a heavy upper tail. After applying the logarithmic transformation, the distribution becomes more compact and variance is reduced, resulting in a more stable learning space. However, the transformation does not eliminate the heavy-tailed nature of the data. Extreme values remain present, although in compressed form, which explains why rare extreme-job prediction remains challenging even after transformation.

4.2. Feature Engineering and Preprocessing

The prediction pipeline is formulated as a sequential learning problem in which each sample is constructed using only past observations to avoid future leakage. The input representation includes raw lag features, logarithmic lag features, rolling statistics, exponential moving averages (EMA), local range indicators, relative ratio features, and local deviation indicators.



(a)



(b)

Figure 2. Distribution of raw and log-transformed cloud job length, (**a**) Raw cloud job-length distribution showing strong right skewness and a heavy upper tail; (**b**) Log-transformed distribution showing a more compact representation.

Table 2. Summary of feature groups and preprocessing configuration

Feature / Preprocessing Component	Count	Description	Configuration
Raw lag features	30	Previous raw cloud job-length observations used as sequential inputs	lag_1 to lag_30
Log-lag features	30	Log-transformed lag observations to reduce skewness	log_lag_1 to log_lag_30, log(1 + MI)
Rolling statistics	40	Local summary statistics of recent temporal behavior	Mean, std, min, max, median, q25, q75, q90 (windows: 3, 5, 10, 20, 30)
Exponential moving average (EMA)	5	Smoothed trend indicators emphasizing recent observations	Spans: 3, 5, 10, 20, 30
Local range features	3	Short-term variability indicators	range_5, range_10, range_20
Ratio, deviation, and spike indicators	9	Relative-change and anomaly-sensitive features	Ratios, zscore_5, zscore_10, spike indicators
Difference features	5	First-order local change indicators (raw and log space)	diff_1, diff_2, diff_3, log_diff_1, log_diff_2
Sequential index feature	1	Positional index of job order	job_index
Target transformation	–	Stabilization of variance under heavy-tailed distribution	$z_t = \ln(1 + y_t)$
Regime threshold	–	Separation into normal and extreme regimes	Quantile (q = 0.80), threshold = 121,000 MI
Sample weighting	–	Emphasis on rare large jobs	Normal = 1.0, Extreme = 2.0, Very Extreme = 3.0

More specifically, the feature set is derived from historical job-length observations and summarizes local temporal behavior through rolling mean, rolling standard deviation, rolling minimum, rolling maximum, and additional distribution-based statistics. EMA features are included to capture smoothed short-term trends, while ratio and deviation-based features are used to detect abrupt workload changes. To mitigate the influence of extreme values, the target variable y_t is transformed using $z_t = \ln(1 + y_t)$. Regime construction is then applied on the training set by defining an extreme threshold based on the upper quantile of the target distribution. Samples above this threshold are assigned to the extreme regime, while the remaining samples belong to the normal regime. In addition, weighted learning is introduced so that extreme and very extreme observations receive higher importance during training. The engineered feature groups and preprocessing configuration are summarized in Table 2. The resulting input representation integrates sequential lags, log-space transformations, statistical summaries, and anomaly-sensitive features. Combined with logarithmic target transformation, regime separation, and weighted learning, this design aims to improve robustness under heavy-tailed cloud job-length variation.

4.3. Baselines and Model Configuration

To ensure a fair evaluation of the proposed framework, the experiments include both simple sequential baselines and stronger machine learning baselines. The sequential baselines consist of Naive_Last and RollingMean_5, while the machine learning baselines include Global_Log_ExtraTrees, RandomForest_Log, GradientBoosting_Log, and MLP_Log. The proposed method, denoted as Regime_Log_ExtraTrees, integrates a regime classifier with separate regressors for normal and extreme samples in the log-transformed space. The inclusion of MLP_Log serves as a neural baseline to provide a basic deep-learning-oriented comparison. However, under the current tabular temporal feature setting, the MLP model does not generalize as effectively as the tree-based ensemble models. This observation indicates that deeper architectures do not necessarily outperform ensemble methods in heavy-tailed cloud job-length prediction when feature representation is limited to engineered tabular inputs.

The emphasis on tree-based models is motivated by the characteristics of both the dataset and the input representation. The experiments are conducted on a relatively small ordered sequence of 1000 jobs, using a tabular feature-engineering design based on lags, rolling statistics, EMA features, and local indicators. In such settings, tree ensembles are typically robust, stable, and less sensitive to hyperparameter tuning compared to deeper sequential architectures. Although sequence-based models such as LSTM or GRU are relevant in broader workload forecasting research, they are not included in the present study because the current framework is formulated around tabular historical features rather than raw sequence tensors. A fair comparison with LSTM/GRU would require a different sequence representation and a dedicated tuning protocol, which is left for future work. The final model families and selected configurations are summarized in Table 3.

Table 3. Final model summary and selected configurations

Model	Family	Target Space	Selected Configuration
Naive_Last	Heuristic	Raw	Uses lag_1 as the next-value predictor
RollingMean_5	Heuristic	Raw	Uses roll_mean_5 as the next-value predictor
Global_Log_ ExtraTrees	Tree ensemble	Log	ExtraTrees with tuned weights (Q = 0.95, Ew = 1.5, VEw = 3.0)
RandomForest_Log	Tree ensemble	Log	RandomForest baseline with log-space target and sample weighting
GradientBoosting_ Log	Tree ensemble	Log	GradientBoosting baseline with log-space target and sample weighting
MLP_Log	Neural baseline	Log	StandardScaler + MLPRegressor in log space
Regime_Log_ ExtraTrees	Hybrid regime-aware model	Log	Classifier + regime-specific regressors (q = 0.80, Ew = 2.0, VEw = 3.0, mode = soft, α = 0.8, threshold = 121,000 MI)

The global and regime-aware configurations were selected based on the best validation performance among the tuning settings explored during experimentation. The selected parameters are Q = 0.95, Ew = 1.5, and VEw = 3.0 for the global model, and q = 0.80, Ew = 2.0, VEw = 3.0, mode = soft, and $\alpha = 0.8$ for the regime-aware model.

4.4. Evaluation Protocol

Model performance is evaluated using RMSE, MAE, MedAE, R^2, and RMSLE. RMSE is particularly important in this study because it penalizes large errors more strongly, which is critical under heavy-tailed job-length behavior. MAE and MedAE capture average and median absolute error, RMSLE reflects relative error in log space, and R^2 measures overall explained variance. The primary evaluation follows a chronological train–validation–test split, ensuring that all predictions are generated using only past observations. This setup preserves the temporal structure of the problem and provides a more realistic evaluation compared to random shuffling.

To further strengthen the evaluation, a walk-forward validation procedure is conducted across five temporal folds. In each fold, the model is trained on earlier data and evaluated on the immediately subsequent segment. This design assesses temporal stability and mitigates the risk of over-reliance on a single split. In addition, ablation experiments are performed to isolate the contribution of regime-aware learning, weighted training, and selected feature groups. Statistical significance analysis is also conducted on fold-wise results to determine whether the observed performance differences are consistent and meaningful rather than incidental.

5. Results and Discussion

5.1. Main Test-Set Comparison

Table 4 summarizes the performance of all models on the chronological test set. The proposed Regime_Log_ExtraTrees achieves the best RMSE of 206,255.66 and the least negative R^2 of -0.01062, indicating the strongest robustness against large prediction errors among the evaluated models. This result is particularly important given the heavy-tailed nature of the target distribution, where rare high-MI jobs dominate operational impact. Under such conditions, RMSE is a critical metric because it penalizes large deviations more strongly than average-error measures. At the same time, the results reveal a clear metric trade-off. While the proposed model performs best in terms of RMSE and R^2, the Global_Log_ExtraTrees baseline achieves the lowest MAE, MedAE, and RMSLE. This indicates that the global log-transformed model remains more competitive in average-case and relative-error performance, whereas the proposed regime-aware framework is more effective in controlling large-error events. Therefore, the contribution of the proposed method should be interpreted as an improvement in robustness under heavy-tailed conditions rather than universal superiority across all evaluation metrics. The additional MLP_Log baseline performs substantially worse than the tree-based models, suggesting that a simple neural architecture does not generalize well under the current tabular temporal feature setting. This observation further supports the practical suitability of tree-based ensembles for the given dataset and feature representation.

Table 4. Performance comparison on the chronological test set

Model	RMSE	MAE	MedAE
Regime_Log_ExtraTrees	**206,255.66**	90,892.53	29,955.43
GradientBoosting_Log	206,639.67	95,887.35	31,977.34
RandomForest_Log	209,560.53	90,299.88	27,042.26
Global_Log_ExtraTrees	211,687.87	**86,877.54**	**26,382.30**
RollingMean_5	226,566.85	113,893.26	52,641.81
Naive_Last	292,561.57	152,690.99	90,242.41
MLP_Log	2,119,819.74	502,816.38	454,334.75

5.2. Walk-Forward Validation

To verify that the observed improvements are not dependent on a single favorable split, the models are further evaluated using five-fold walk-forward validation. The proposed

Regime_Log_ExtraTrees achieves the best mean RMSE of 174,889.33 ± 41,129.27 and the best mean R^2 of −0.03140 ± 0.03079 across temporal folds. The closest competitor is Global_Log_ExtraTrees, which achieves a mean RMSE of 176,946.47 ± 44,119.23 and a mean R^2 of −0.04729 ± 0.02728.

These results reinforce the main finding of this study. The regime-aware model consistently maintains the strongest robustness as the evaluation window shifts over time, indicating that its advantage is not limited to a specific data partition. However, the performance margin relative to strong tree-based baselines remains modest. This suggests that regime-aware learning should be interpreted as a targeted improvement for heavy-tailed conditions, rather than a universally dominant modeling approach. The aggregated walk-forward validation results are presented in Table 5.

Table 5. Walk-forward validation results

Model	RMSE (mean ± std)	MAE (mean ± std)	MedAE (mean ± std)	R^2 (mean ± std)	RMSLE (mean ± std)
Regime_Log_ExtraTrees	**174,889.33 ± 41,129.27**	85,739.14 ± 13,501.99	45,919.87 ± 2,301.69	**−0.03140 ± 0.03079**	0.74909 ± 0.03187
RandomForest_Log	176,365.12 ± 43,915.24	88,608.97 ± 45,597.50	48,658.46 ± 45,636.49	−0.04180 ± 0.03920	0.74764 ± 0.17185
Global_Log_ExtraTrees	176,946.47 ± 44,119.23	**71,468.17 ± 15,710.99**	**26,320.59 ± 2,080.85**	−0.04729 ± 0.02728	**0.68679 ± 0.06874**
GradientBoosting_Log	179,994.54 ± 48,482.02	96,220.39 ± 49,800.90	55,282.09 ± 53,194.63	−0.07780 ± 0.10236	0.78937 ± 0.18467
RollingMean_5	188,705.48 ± 46,188.13	105,501.65 ± 28,573.54	45,600.00 ± 9,362.96	−0.19595 ± 0.04466	0.86940 ± 0.11103
Naive_Last	248,329.79 ± 61,620.70	121,298.97 ± 28,697.18	36,800.00 ± 4,604.35	−1.06994 ± 0.13678	0.98479 ± 0.08418

For completeness, the fold-wise RMSE and R^2 values of the proposed Regime_Log_ExtraTrees model are reported in Table 6.

Table 6. Fold-wise RMSE and R^2 for Regime_Log_ExtraTrees.

Fold	RMSE	R^2
1	106,154.08	−0.07639
2	182,246.23	−0.03326
3	216,223.91	−0.02500
4	178,650.85	−0.03242
5	191,171.57	0.01010

5.3. Ablation Study

An ablation study was conducted to assess the contribution of the main components of the proposed framework. The results indicate that regime-aware modeling consistently contributes to competitive performance, whereas the effects of weighted learning and the ratio/spike/z-score feature subset are more variable. In particular, the variant without the ratio/spike/z-score subset achieves a slightly improved RMSE of 205,914.24, while the regime-based model without weighting remains highly competitive with an RMSE of 205,992.81. By comparison, the full model achieves an RMSE of 206,255.66. These findings suggest that the primary strength of the proposed framework lies in regime separation, rather than in the cumulative contribution of all auxiliary components.

Weighted learning and additional extreme-sensitive features may still provide benefits in certain cases; however, their impact is not consistently positive across all evaluation metrics in the current dataset. This observation is important because it demonstrates that the framework is not merely an aggregation of feature enhancements. Instead, its main effectiveness arises from explicitly distinguishing between normal and extreme job-length patterns. The ablation variants are summarized in Table 7.

Table 7. Ablation study of regime modeling, weighting, and feature subsets.

Variant	RMSE	MAE	MedAE	R²	RMSLE
A5_FullMinusRatioSpikeZscore	**205,914.24**	96,809.38	44,681.48	−0.00728	0.79006
A3_Regime_NoWeight	205,992.81	96,152.28	40,479.47	−0.00805	0.78837
A4_Full_Regime_Weighted	206,255.66	95,635.37	37,613.92	−0.01062	0.78642
A0_Raw_ExtraTrees	206,552.51	105,896.30	53,024.06	−0.01353	0.81904
A2_LogPlusWeight_Global	211,687.94	86,877.54	26,382.30	−0.06456	0.76504
A1_LogOnly_ExtraTrees	211,894.05	86,841.71	26,633.37	−0.06663	0.76485

5.4. Error Analysis on Extreme Jobs

Figure 3 presents the actual and predicted cloud job length on the chronological test set for the proposed Regime_Log_ExtraTrees model and the Global_Log_ExtraTrees baseline. Both models capture the central tendency of the series reasonably well, particularly in the dominant mid-range region. However, neither model reproduces the full amplitude of the largest spikes.

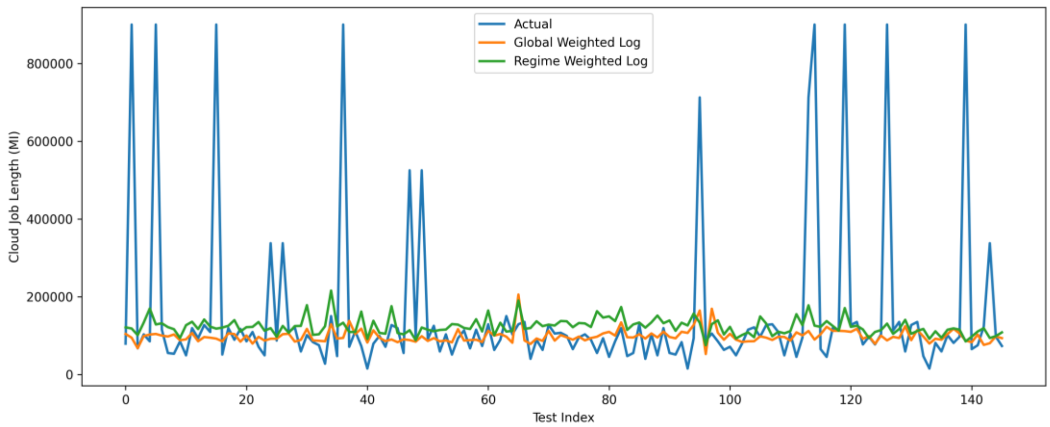


Figure 3. Actual and predicted cloud job length on the chronological test set.

The regime-aware model shows slightly stronger responses in several high-value regions compared to the global baseline, but substantial underestimation remains for the most extreme jobs. This behavior is further illustrated in Figure 4, where many high-MI observations are mapped to significantly lower predicted values, indicating a systematic compression of the upper tail.

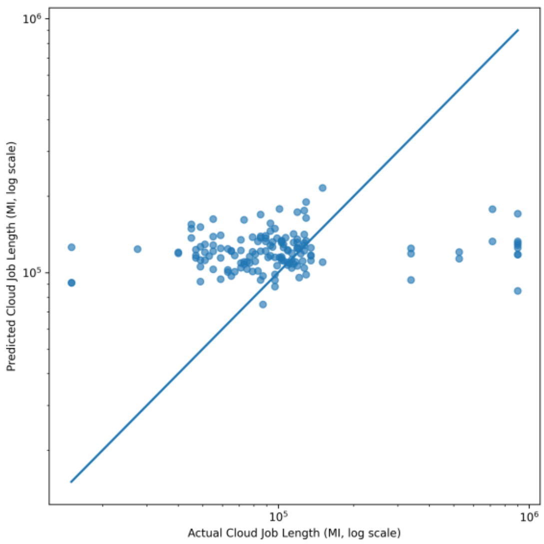


Figure 4. Scatter plot of actual versus predicted cloud job length in logarithmic scale.

From a practical perspective, the proposed framework improves robustness against large prediction errors but does not fully resolve the challenge of rare extreme-job prediction. This limitation is likely attributable not only to the heavy-tailed nature of the target distribution but also to the restricted feature representation. The current framework relies primarily on historical job-length dynamics and does not incorporate richer contextual variables such as resource requests, workload categories, or job-level metadata.

5.5. Statistical Significance and Overall Discussion

Table 8 reports the statistical significance analysis based on fold-wise comparisons between the proposed model and competing baselines. The results show that the proposed method achieves statistically significant improvements over weaker baselines such as RollingMean_5 and MLP_Log. However, the differences between Regime_Log_ExtraTrees and stronger tree-based models are smaller and are not consistently significant across all comparisons. This finding aligns with the numerical results: while the proposed method improves robustness, the performance gain over the best non-regime ensemble baselines remains moderate.

Another important observation is that all evaluated models produce negative R^2 values, indicating that cloud job-length prediction under the present heavy-tailed setting remains intrinsically difficult. Under such conditions, R^2 should be interpreted cautiously, and greater emphasis should be placed on robustness-oriented metrics such as RMSE. Therefore, the contribution of this study should not be interpreted as achieving perfect predictive accuracy, but rather as providing a more robust strategy for controlling large prediction errors under severe target skewness. In this context, the proposed method offers a practical improvement over simple sequential baselines and remains competitive with strong global ensemble models.

Table 8. Statistical significance test results on fold-wise performance.

Comparison	Two-sided p-value	One-sided p-value (Regime better)	Interpretation ($\alpha = 0.05$)
Regime vs Naive_Last	0.77356	0.61322	Not significant
Regime vs RollingMean_5	1.38×10^{-4}	6.90×10^{-5}	Significant; supports Regime_Log_ExtraTrees
Regime vs Global_Log_ExtraTrees	8.35×10^{-24}	1.00000	Significant difference, not in favor of Regime
Regime vs RandomForest_Log	0.46535	0.76732	Not significant
Regime vs GradientBoosting_Log	0.46694	0.23347	Not significant
Regime vs MLP_Log	1.49×10^{-10}	7.44×10^{-11}	Significant; supports Regime_Log_ExtraTrees

A remaining limitation is that this study does not include a full like-for-like reimplementation of external state-of-the-art methods on the same GoCJ-derived setting. Prior studies often differ in prediction targets, feature representations, and evaluation protocols, which makes direct numerical comparison difficult. Therefore, the reported results should be interpreted as a controlled internal benchmark. Future work should include stricter same-dataset comparisons under a unified experimental protocol.

6. Conclusions

This study proposed a log-transformed, regime-based machine learning framework for cloud job-length prediction under heavy-tailed workload conditions. The framework integrates sequential feature engineering, logarithmic target transformation, weighted learning, and regime-aware modeling to explicitly separate normal and extreme job-length patterns. Experimental results on an ordered GoCJ-derived job-length sequence show that the proposed Regime_Log_ExtraTrees model achieves the best RMSE and the least negative R^2 on the main chronological test set, indicating improved robustness against large prediction errors compared to the evaluated baselines. Additional walk-forward validation further confirms that the regime-aware model consistently achieves the best mean RMSE and mean R^2 across temporal folds, suggesting that its advantage is not limited to a single data partition.

The ablation study demonstrates that regime separation is the primary contributor to robustness, while weighted learning and certain extreme-sensitive feature subsets yield more variable effects. Error analysis also reveals that the proposed method still underestimates the largest spikes, indicating that accurate prediction of rare extreme jobs remains an open challenge. Therefore, the contribution of this work should not be interpreted as achieving universally superior predictive accuracy, but rather as demonstrating that log-transformed, regime-aware learning provides a practical and more robust strategy for cloud job-length prediction under strong skewness and heavy-tailed variation. Future work may focus on incorporating richer contextual features, exploring uncertainty-aware prediction, and extending evaluation to multiple datasets in order to improve generalizability and better capture extreme-tail behavior.

Author Contributions: Conceptualization: A.P, B.R. and F.F.R.; Methodology: A.P.; Software: B.R.; Validation: A.P., B.R. and F.F.R.; Formal analysis: A.P.; Investigation: B.R.; Resources: A.P.; Data curation: F.F.R.; Writing—original draft preparation: A.P.; Writing-review and editing: A.P.; Visualization: B.R.; Supervision: F.F.R.; Project administration: F.F.R.; funding ac-quisition: B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The dataset used in this study was derived from the publicly available GoCJ (Google Cloud Jobs) benchmark dataset for distributed and cloud computing research. The dataset is archived in the Mendeley Data repository and is described in Hussain and Aleem [31]. In this study, an ordered sequence of 1000 job-length records was extracted and preprocessed for sequential modeling. The dataset can be accessed at: https://data.mendeley.com/datasets/b7bp6xhrcd/1

Acknowledgments: The authors would like to thank Universitas Ahmad Dahlan for institutional support.

Conflicts of Interest: The authors declare no conflict of interest

References

[1] S. Kashyap, A. Singh, and S. S. Gill, "Machine learning-centric prediction and decision based resource management in cloud computing environments," *Cluster Comput.*, vol. 28, no. 2, p. 130, Apr. 2025, doi: 10.1007/s10586-024-04787-8.

[2] A. Rossi, A. Visentin, D. Carraro, S. Prestwich, and K. N. Brown, "Forecasting workload in cloud computing: towards uncertainty-aware predictions and transfer learning," *Cluster Comput.*, vol. 28, no. 4, p. 258, Aug. 2025, doi: 10.1007/s10586-024-04933-2.

[3] S. Kashyap and A. Singh, "Prediction-based scheduling techniques for cloud data center's workload: a systematic review," *Cluster Comput.*, vol. 26, no. 5, pp. 3209–3235, Oct. 2023, doi: 10.1007/s10586-023-04024-8.

[4] 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, pp. 6481–6490, Sep. 2022, doi: 10.1016/j.jksuci.2021.04.011.

[5] T. Ali, H. U. Khan, F. K. Alarfaj, and M. AlReshoodi, "Hybrid deep learning and evolutionary algorithms for accurate cloud workload prediction," *Computing*, vol. 106, no. 12, pp. 3905–3944, Dec. 2024, doi: 10.1007/s00607-024-01340-8.

[6] M. Xu, C. Song, H. Wu, S. S. Gill, K. Ye, and C. Xu, "esDNN: Deep Neural Network Based Multivariate Workload Prediction in Cloud Computing Environments," *ACM Trans. Internet Technol.*, vol. 22, no. 3, pp. 1–24, Aug. 2022, doi: 10.1145/3524114.

[7] J. Kumar and A. K. Singh, "Performance Assessment of Time Series Forecasting Models for Cloud Datacenter Networks' Workload Prediction," *Wirel. Pers. Commun.*, vol. 116, no. 3, pp. 1949–1969, Feb. 2021, doi: 10.1007/s11277-020-07773-6.

[8] M. E. Karim, M. M. S. Maswood, S. Das, and A. G. Alharbi, "BHyPreC: A Novel Bi-LSTM Based Hybrid Recurrent Neural Network Model to Predict the CPU Workload of Cloud Virtual Machine," *IEEE Access*, vol. 9, pp. 131476–131495, 2021, doi: 10.1109/ACCESS.2021.3113714.

[9] Y. S. Patel and J. Bedi, "MAG-D: A multivariate attention network based approach for cloud workload forecasting," *Futur. Gener. Comput. Syst.*, vol. 142, pp. 376–392, May 2023, doi: 10.1016/j.future.2023.01.002.

[10] 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, Aug. 2021, doi: 10.1007/s00521-021-05770-9.

[11] L. Kidane, P. Townend, T. Metsch, and E. Elmroth, "Automated Hyperparameter Tuning for Adaptive Cloud Workload Prediction," in *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, Dec. 2023, pp. 1–8. doi: 10.1145/3603166.3632244.

[12] J. Bawa, K. Kaur Chahal, and K. Kaur, "Improving cloud resource management: an ensemble learning approach for workload prediction," *J. Supercomput.*, vol. 81, no. 10, p. 1138, Jul. 2025, doi: 10.1007/s11227-025-07560-9.
[13] T. Khan, W. Tian, S. Ilager, and R. Buyya, "Workload forecasting and energy state estimation in cloud data centres: ML-centric approach," *Futur. Gener. Comput. Syst.*, vol. 128, pp. 320–332, Mar. 2022, doi: 10.1016/j.future.2021.10.019.
[14] J. Lwin, "Enhancing Cloud Task Scheduling with Multi-Objective Optimization Using K-Means Clustering and Dynamic Resource Allocation," *J. Comput. Theor. Appl.*, vol. 2, no. 2, pp. 202–211, Oct. 2024, doi: 10.62411/jcta.11337.
[15] N. K. Mon, "Optimizing Cloud Computing Performance by Integrating the Novel PSBR Service Broker Policy and Load Balancing Algorithms," *J. Comput. Theor. Appl.*, vol. 2, no. 2, pp. 212–221, Oct. 2024, doi: 10.62411/jcta.11221.
[16] A. Bin Faisal, N. Martin, H. M. Bashir, S. Lamelas, and F. R. Dogar, "When will my ML Job finish? Toward providing Completion Time Estimates through Predictability-Centric Scheduling," in *Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, 2024, pp. 487–505. doi: 10.5555/3691938.3691964.
[17] Y. Xing, "Work Scheduling in Cloud Network Based on Deep Q-LSTM Models for Efficient Resource Utilization," *J. Grid Comput.*, vol. 22, no. 1, p. 36, Mar. 2024, doi: 10.1007/s10723-024-09746-6.
[18] S. Lajili, Z. Brahmi, and M. N. Omri, "ML WPStreamCloud: ML-based Workload Prediction and Task Clustering for Efficient Stream Application Ofoading in Heterogeneous Edge and Cloud Environments," *Procedia Comput. Sci.*, vol. 246, pp. 1527–1537, 2024, doi: 10.1016/j.procs.2024.09.610.
[19] E. Yildirim, M. Hussein, M. Titov, and O. O. Kilic, "Predicting runtime and resource utilization of jobs on integrated cloud and HPC systems," *Futur. Gener. Comput. Syst.*, vol. 176, p. 108230, Mar. 2026, doi: 10.1016/j.future.2025.108230.
[20] N. I. Mahbub, M. D. Hossain, S. Akhter, M. I. Hossain, K. Jeong, and E.-N. Huh, "Robustness of Workload Forecasting Models in Cloud Data Centers: A White-Box Adversarial Attack Perspective," *IEEE Access*, vol. 12, pp. 55248–55263, 2024, doi: 10.1109/ACCESS.2024.3385863.
[21] P. Nehra and N. Kesswani, "A workload prediction model for reducing service level agreement violations in cloud data centers," *Decis. Anal. J.*, vol. 11, p. 100463, Jun. 2024, doi: 10.1016/j.dajour.2024.100463.
[22] H. L. Leka, Z. Fengli, A. T. Kenea, A. T. Tegene, P. Atandoh, and N. W. Hundera, "A Hybrid CNN-LSTM Model for Virtual Machine Workload Forecasting in Cloud Data Center," in *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Dec. 2021, pp. 474–478. doi: 10.1109/ICCWAMTIP53232.2021.9674067.
[23] K. Menear, A. Nag, J. Perr-Sauer, M. Lunacek, K. Potter, and D. Duplyakin, "Mastering HPC Runtime Prediction: From Observing Patterns to a Methodological Approach," in *Practice and Experience in Advanced Research Computing*, Jul. 2023, pp. 75–85. doi: 10.1145/3569951.3593598.
[24] Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, A. Basuhail, and K. Jambi, "Deep Reinforcement Learning for Workload Prediction in Federated Cloud Environments," *Sensors*, vol. 23, no. 15, p. 6911, Aug. 2023, doi: 10.3390/s23156911.
[25] S. Karimunnisa and Y. Pachipala, "Deep Learning Approach for Workload Prediction and Balancing in Cloud Computing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 4, 2024, doi: 10.14569/IJACSA.2024.0150477.
[26] O. Ghandour, S. El Kafhali, and M. Hanini, "Adaptive workload management in cloud computing for service level agreements compliance and resource optimization," *Comput. Electr. Eng.*, vol. 120, p. 109712, Dec. 2024, doi: 10.1016/j.compeleceng.2024.109712.
[27] S. Kashyap and A. Singh, "A Hybrid Scheduling for Multi-Objective Optimization using Prediction Approach," *J. Grid Comput.*, vol. 23, no. 3, p. 22, Sep. 2025, doi: 10.1007/s10723-025-09809-2.
[28] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions," *Artif. Intell. Rev.*, vol. 57, no. 5, p. 124, Apr. 2024, doi: 10.1007/s10462-024-10756-9.
[29] Y. Sanjalawe, S. Al-E'mari, S. Fraihat, and S. Makhadmeh, "AI-driven job scheduling in cloud computing: a comprehensive review," *Artif. Intell. Rev.*, vol. 58, no. 7, p. 197, Apr. 2025, doi: 10.1007/s10462-025-11208-8.
[30] K. Menear, K. Konate, K. Potter, and D. Duplyakin, "Tandem Predictions for HPC jobs," in *Practice and Experience in Advanced Research Computing 2024: Human Powered Computing*, Jul. 2024, pp. 1–9. doi: 10.1145/3626203.3670547.
[31] A. Hussain and M. Aleem, "GoCJ: Google Cloud Jobs Dataset for Distributed and Cloud Computing Infrastructures," *Data*, vol. 3, no. 4, p. 38, Sep. 2018, doi: 10.3390/data3040038.