

An Intelligent Learning Model For Workload Prediction In Cloud Environment

Yaddala Srinivasulu, Research Scholar, Department of Computer Science Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.

²Bobba Basaveswara Rao, Department of Computer Science Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.

Dr.G.Neelima, ³Assistant Professor, Department of Computer Science & Engineering, ANU College of Sciences, Acharya Nagarjuna University.

Abstract— Cloud computing is the domain that can be used by the many organizations to provide the dynamic and accurate application that can work on heterogeneous workloads. This study proposed a Intelligent Ensemble Learning Model (ILM) which is used to predict the dynamic workload in cloud environments. The proposed model analyzes both historical and real-time workload characteristics such as CPU utilization, memory consumption, task arrival rate/ execution time and resource demand to look for fluctuations and temporal patterns. First, raw workload data are preprocessed by means of normalization, noise reduction and feature transformation to enhance the prediction consistency. Then, concurrently features-related and temporal adaptation is learned with a multitude of Novel Temporal Workload Predictor (NTWP) whose outputs are ensemble together adapted. By taking advantage of the complementary strengths of individual learners, the ensemble strategy leads to more robust predictions and helps mitigate prediction errors in highly dynamic workloads. The workload information prediction could later on help with proactive resource provisioning and dynamic cloud resource management. The experiments are conducted on Alibaba Cluster Trace 2018 dataset that process the data from 4k machines. Finally, the workload prediction utilized for proactive resource provisioning as well as instant cloud provision. Results show that the proposed approach achieved the superior performance in terms of effective resource usage in cloud platforms.

Keywords— Cloud Computing, Load Balancing, Novel Temporal Workload Predictor (NTWP), Dynamic Work Loading, Alibaba Cluster Trace 2018 dataset.

I. INTRODUCTION

Cloud Computing is a fast evolving field and it has led to very dynamic workloads created with web applications, enterprise services, IoT platforms, Data Analytics and Artificial Intelligence applications [1]. Such workloads execute repeatedly, changing continuously CPU utilization, memory consumption, I/O (network traffic and storage), request arrival rates, and duration of execution which often complicate the management of cloud resources [2]. Accurate Workload Prediction allows cloud providers to allocate virtual machines, containers, processing capacity and other resources in anticipation of future resource requirements. Predicting the workload variation in advance, cloud systems reduce underutilization of resources, avoid performance degradation, increase SLA compliance and save operational costs [3].

In terms of traditional workload prediction approaches, they typically seek to find that there are statistical properties instead of the property of utilization along with historical cloud resource usage using conventional machine-learning algorithms [4]. Our cloud workloads are nonlinear, highly dynamic, seasonal and temporal so conventional approaches may provide challenges to the predictive modeling capability. Another advantage of deep learning is that it allows us to automatically learn the complex temporal dependencies from past workload observations. Several models like LSTM, GRU, CNN, TCN, Transformers and hybrid DL architectures to memorize short-term and long-time workloads [5]. These workloads can better learn the amount of CPU, memory, network, requests and resource-utilization information from past experience to predict future workload levels [6]. An effective cloud workload prediction framework can combine advanced preprocessing, temporal feature extraction, deep-learning-based prediction and adaptive resource-management mechanisms [7]. A

framework of this kind will be able continuously to monitor past and present workload data, recognize changes in utilization patterns, and ultimately predict cloud resource needs well ahead of time [8]. Such peak workload prediction then can facilitate dynamic resource provisioning, virtual-machine scaling, workload scheduling, load balancing, energy optimization and SLA-aware resource allocation. Thus, intelligent workload prediction is a key element of self-organizing clouds and lifts clouds more towards adaptive, cost-effective, scalable and responsive behavior to changing application demand.

II. LITERATURE SUREVY

Paul et al. [10] introduced the PS-ARO hybrid optimization algorithm for task scheduling and resource allocation in cloud environments. It is a novel hybrid optimization algorithm which with PS based this work integrates ARO to combine usage into mechanism PS-ARO. Firstly, the proposed approach models the scheduling problem using information about jobs and job characteristics such as virtual-machine capacity, Execution requirements, and availability of resources. Pattern Search is used for systematic local search of promising scheduling solutions while Artificial Rabbit Optimization employs dynamic search to improve the task-to-resource assignment in both global exploration and global exploitation. The complementary Nature of PS and ARO mitigates the problem of premature convergence that enhances the capacity to discover better scheduling decision in a more complex search space. Zhuang et al. [11] focuses on FAS, which we call instance seeking CD, investigates the influence of a set of instance seeking strategies to allocate resources in cloud data centers with the goal of matching available cloud resources with computational workloads. Important resource-management factors such as CPU and memory utilization, execution time, energy consumption, load balancing efficiency, resource allocation efficiency response time and cost are all considered in the analysis. Based on that, various instance selection strategies can be evaluated in different workload scenarios to assess their ability to cope with dynamic and heterogeneous cloud environments. Additionally, intelligent methods utilizing machine learning, predictive analytics and metaheuristic optimization can optimize instance selection by anticipating workload needs and dynamically discovering ideal resources.

Sayegh et al. [12] proposes a Mobility-Aware Task Migration approach for VEC networks that decides if, when and where an active task should migrate based on not only patterns of vehicle movement, requirements during the process of task execution as well as availability of edge-servers and communication conditions but also how long the connection between vehicles can sustained. Our solution constantly observes the mobility state of the vehicles, as well as the resource condition of nearby edge nodes to a timely find optimal migration targets before its current connection is no longer available. Joint optimization of migration cost, task completion time, communication latency, availability of computational resource, energy consumption and service continuity makes the task migration optimal. The framework combines mobility prediction with intelligent task migration and resource allocation to minimize the failure of tasks, however, effective migration operations also contribute to better edge-resource utilization and application responsiveness. Zhanuzak. [13] survey explores various significant dynamic load-balancing strategies like centralized, distributed, hierarchical, heuristic-based, metaheuristic based and machine learning. Focus is placed on their effectiveness in responding to key issues - imbalance in workload distribution, underutilization of resources, response time, makespan, energy usage, communication overheads and SLA violations. The review also examines smart methods that enhances adaptive task allocation through the study of different workload stream patterns, and then automatically assign to available suitable resources. This brings a discussion of related work on how these approaches trade off computational workload, decision accuracy, scalability and efficiency of resources.

Alsheavi et al. [14] proposed an hybrid load-balancing approach with ACO and CSA is introduced for task allocation and workloads distribution in cloud computing environment. ACO has pheromone-based learning to anticipate possible resource-allocation paths WARD uses a popular clustering method called component face where the behavior and likelihood of each node are formalized in a Boolean Matrix, then above ACO for resource allocation; CSA uses intelligent exploration/exploitation inspired by the crows food-searching and hiding strategies. The overall optimization strategy is being proposed to discover suitable task-to-resource mappings while aiming to minimize or maximize important cloud performance metrics, e.g. Makespan, Response Time, Resource Utilization, Energy Cost, Load Balance and Execution Cost. All in all, hybrid

approach is aimed at escaping premature convergence as well as the ideal searching for efficient resource allocations of complex cloud environment. Shafiq et al. [15] reviews the significant static and dynamic centralized, decentralized load-balancing methods in cloud computing and heuristic methods, metaheuristic optimization algorithms and machine-learning-based techniques. The most notable performance parameters considered in the survey include makespan, response time, execution time to complete a job, resource utilization (which is important for workload consolidation), energy consumption, load imbalance as well as throughput and cost. It further reflects upon issues such as treating heterogeneous resources and unpredictable workload, large scale cloud infrastructures with dynamic resource conditions which limits the traditional approaches.

Kaviarasan et al. [16] propose a highly effective loadbalancing approach that efficiently determines assignments of task to resource using an Inspired Lion Optimization Algorithm (ILOA) based on dynamic cloud environments. The optimization strategy proposed takes advantage of the characteristics of lions behaviour to be able to search and exploit the cloud resource search space in order to find optimal resource allocations appropriate for a situation defined by workload and resources conditions. Analyzing the execution of the proposed ILOA based algorithm with an dynamic workload to determine if it out performs commonly used and accepted metaheuristic load-balancing algorithms. Mishra et al. [17] gives a broad view about load balancing in cloud computing by an exploration of the fundamental ideas, aims as well as techniques and architectures to strategically distribute workloads across multiple resources. The work presented here opens up a number of effective potential routes to further reduce the dependence on human skill and availability in adaptive load-balancing systems. Abstract In this paper, an overview of existing techniques and state-of-the-art intelligent approaches enables us to consider adaptive, scalable, energy-efficient and workload-aware load balancing criteria for building efficient cloud computing environments with reliability and high-performance.

Vijay et al. [18] introduced the cloud computing algorithms provide effective mapping of user asking to appropriate virtual machine and physical resources. It also highlights various issues related to heterogeneous cloud resources, uncertainty of workloads in real-time environments, scalability, task migration and communication overhead and resource availability. The Integration of resource scheduling and load balancing can immensely enhance Cloud Performance, Scalability, Resource utilization Efficiency, Energy Efficiency and Service Reliability establishing a backbone for intelligent and adaptive cloud computing environments. Raghav et al. [19] survey work of dynamic load-balancing techniques for cloud environment are survey and the centralised, distributed, hierarchical along with heuristic, metaheuristics and ML-based approaches. The survey points challenges on unpredictable workload, heterogeneous resources, communication overhead, task migration and resource contention and the challenge of large-scale cloud infrastructures. A comparative study across existing methods indicates trade-offs between complexity, decision-making accuracy, scalability, and resource efficiency. Kaviarasan et al. [20] presented the load balancing approach based on EMAO-MBO is represented in [20]. The idea behind the proposed approach is to utilize a global exploration capability of MBO while enhancing migration and adjustment operations similar to those in ACO (ants here represent candidates), which can enhance search efficiency for resource-allocation solutions. The migration operator allows for exploration of different resource configurations, and the adjustment operator refines promising solutions and improves the distribution of workloads. Under varying workload scenarios, performance can be compared with traditional for load-balancing approaches are optimization-based.

I. METHODOLOGY

This paper introduces the Intelligent Learning Model (ILM) for dynamic cloud workload prediction as it consistently learns from both the historical and real-time characteristics of workload patterns. In this framework, basic workload indicators of cloud applications like CPU utilization, memory consumption, task arrival rate, work execution time and resource requirement are considered to describe the behavior of computational aspect of cloud application (Ughrov et al., 2022). First we will describe how to collect workload observations from cloud resources and then introduce the preprocessing step which aims at enhancing data quality and making predictions more consistent. In specific, normalization changes the heterogeneous workload attributes into ranges with smaller differences (e.g., [0.0, 1.0] to return values on similar orders of magnitude), noise reduction removes noise from outliers or unreliable observations, and feature transformation translates raw workload information into a form appropriate for ML models. This processed workload sequence is then fed into the prediction stage proposed in this work.

The second stage of Intelligent Learning Model (ILM) is that it applies several Novel Temporal Workload Predictors (NTWPs) to extract the features and find temporal variations with regard to workloads. Rather than using one prediction model, multiple learners run simultaneously and cover various aspects of workload behaviour (short-term fluctuations, changing resource demand, task-arrival patterns and temporal dependencies). In addition, each NTWP conducts separate workload prediction and the predictions are combined by an ensemble approach. The ensemble methodology takes advantage of the complementary merits of the base learners and diminishes the effect of mistakes made by any individual classifier (predictor). This allows ILM to make more simultaneous workload predictions, even in highly dynamic Situations where workload intensity can change rapidly.

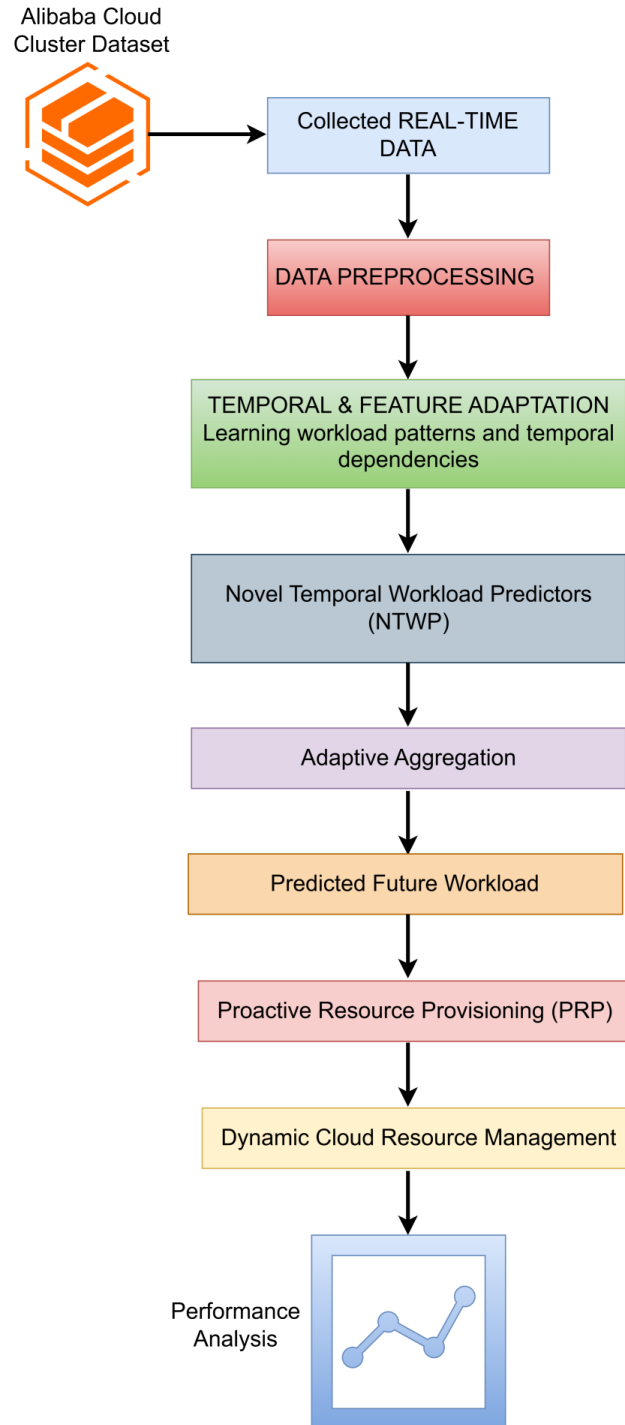


Figure 1: Architecture Diagram for ILM

5.3 Proposed Methodology: Intelligent Learning Model (ILM)**5.3.1 Pre-processing**

In the ILM, they have observed a raw cloud workload from which it obtains cleaned and stable data for workload prediction. These data may include some factors ranging from processor usage, memory usage, and task requesting level, task processing time and resource requirement. Step1: Handle missing/invalid observations in a way that they do not affect badly during model training. It is to be followed by denoise for removing unexpected oscillations and out-of-spec workload readings that can get in the way of good prediction (again, performance). The next step is normalization, which rescales your features so that the features with larger numeric ranges do not dominate and distort the data. Finally, from the input type which you will work on collected workload data in a tabular form in feature transformation transform this into proper shape which can be processed by NTWPs while still taking accounting for time ordering. As a result, we attain a clean and consistent dataset that is then forwarded to the aforementioned temporal feature-learning layer for reliable and even workload prediction.

5.3.2 Temporal & Feature Adaptation

Temporal and Feature Adaptation is a crucial part of the proposed ILM, which learns properties of each workload feature, as well as changes in data over time. The reference data of cloud workload consists of variables such as CPU usage, memory usage, task arrival rate, task execution time and resource demand which have different temporal behaviors in terms of processing. This learns the impacts and their interactions with each other; whereas temporal adaptation empowers the model to capture sequential data associated with short-term changes, long-term patterns, seasonal behavior, and abrupt workload shifts. Higher task arrival rates would then lead to high CPU and memory utilizations, with delays in executing tasks interpreted as a variation of resource requirements. Rather than treating each observation separately, with ILM we learn such relationships between features and temporal dependencies jointly to construct a context-sensitive representation of the state of its current and evolving workload.

Next, the processed representation is inputted into multiple NTWPs where separate learners identify different forms of workload and predict future-workload individually. This adaptation mechanism is important in the case of cloud where workload behavior varies significantly with different parameters such as number of user requests, applications to be executed and resource requirements. The model continues to update its earlier trends in time learnt via previous workloads, thus it is able to adapt itself with the dynamism of workload condition and make future predictions sturdy. As a result, the temporal-feature representation becomes an intermediate information layer between data preprocessing and ensemble prediction, enabling ILM with more precise workload predictions which will later assist it in proactive resource provisioning, dynamic load balancing and adaptive cloud resource management.

5.3.3 Novel Temporal Workload Predictors (NTWP)

The second stage of the proposed novel ILM learns both the characteristics of individual workload features and their time variations. The processed cloud workload data include CPU utilization, memory consumption, task arrival rate, task execution time and resource demand which have different temporal behaviors. Whereas the feature-adaptation module learns the importance and interrelation of these variables, temporal adaptation uses time series observations to model short-term variations, long-term evolution trends, periodic patterns and sudden load shifts. That is, an increase in arrival rate of tasks can lead to higher CPU and memory usage subsequently-maybe changes in execution time reflect variation in the demand for resources. Such that with jointly learning these relationships, ILM learns a contextual model of the workload state and evolution in time instead of viewing each workload observation as separate.

The adapted representation is then passed to multiple NTWPs where distinct learners can discover complementary workload patterns and produce individual future-workload predictions. One of the

most important features to a cloud environment is the adaptation mechanism because workload behavior can vary greatly within short time frames due to fluctuating user requests, application run-time requirements, and resource demands. The model utilizes the management of new workload observations and previously learned temporal patterns to respond rapidly and robustly to changing workloads in order to improve future predictions. Thus, the generated temporal-feature representation would function as the intermediate information layer between data preprocessing and novel prediction stage in ILM; enabling ILM to have better reliable workload forecasting which could later on helped in dynamic load balancing, proactive resource provisioning, and adaptive cloud resource management.

5.3.4 Adaptive Aggregation

Adaptive Aggregation is an element of ILM that merges the departmental workload predictions from a combination of NTWPs into one aggregate, more accurate workload predictor. Each NTWP might learn different facets of cloud workload behavior like short-term CPU variations, memory-consumption characteristics, task-arrival and execution-time patterns or even resource-demand trends. Unlike previous studies which relied on the output of a single NTWP, we collected outputs from multiple NTWPs and aggregated them based on an appropriate strategy (i.e. weighted average prediction or adaptive prediction weighting). The reason being is that she can fashion the contribution of each predictor based on how reliable prediction they yield, so a more accurate learner would be given a greater weight on top of her workload estimation. As a result, this process generates a single prediction that combines the knowledge of the individual temporal predictors.

The second primary role of adaptive aggregation is to mitigate prediction errors and to enhance resilience in an increasingly volatile cloud workload environment. When abrupt changes occur in workload patterns, individual predictors are prone to making errors as they may contain noise or spikes and drops that the predictors did not anticipate. Nonetheless, these errors are likely to be different at the level of individual learners, so aggregating their predictions will average out an inaccurate prediction and generate a more robust final estimate. This reduces the susceptibility to improving upon one single NTWP, while also enhancing prediction consistency across disparate workload conditions. This aggregated workload prediction can be provided as an input to proactive resource provisioning and dynamic cloud resource management, so that the cloud system identifies changing workloads, and adjusts resources proactively such that systems are not overloaded or underutilized and remain performing efficiently.

5.4 Proactive Resource Provisioning (PRP)

The last stage of resource-management in the proposed ILM is Proactive Resource Provisioning, where a more accurate prediction from future workload generation induced by adaptive aggregation is used to provision cloud resources before real changes of workload. Rather than waiting for a demand for CPU, memory, or task to hit an overload condition, the system looks at workload forecasts and determines what resources are likely needed and in what amount. Resource provisioning automatically updates, either scaling up or down computation based on expected activity. During a period where it is expected there will be high-demand on services, additional resources could be spun up and when not required, unnecessary resources can also spin down. This synergy of predictive techniques lowers the probability for resource saturation and helps to reduce unnecessary provisioning of compute capacity and also improves utilization. The capacity of the provisioned resource can be determined by taking task workload characterization (for example, relevant parameters include task arrival rate, execution time, cpu utility and memory consumption) into account.

Depending on the predicted workload, it can choose optimal Virtual Machines (VMs) allocate them to an incoming or anticipated task in relation to their processing needs and availability of resources at any point in time. Resource adjustment is a cloud resource rental method that dynamically adjusts the CPU, memory, or other allocated resources based on its predicted demand and updates the allocation optimally in real-time without fixing assignments for each VM. Workload redistribution transfers (or releases or reassigns) tasks towards resources that are available and less busy VMs or containers to mitigate resource bottlenecks when expectations arise on work concentration of specific resources, in order to preserve balanced utilization. As a result, thanks to proactive provisioning, the workload predictions of ILM are turned into effective resource-management decisions and thus can facilitate quicker response time, better resource utilization, lower overload probability (saving power), elasticity/scalability of cloud systems as well as more stable performance for cloud-systems.

5.5 Dynamic Cloud Resource Management

In this stage the cloud resource management to issue provisioning decision based on the predicted workload performed in real-time through ILM. The system observes present CPU, memory, virtual machine, container and other computational resource utilizations as well as their availabilities then compares against the expected utilization and availability based on prediction of workload requirements. Using this information the resources can be dynamically activated, released, resized or reassigned to change workload conditions. This continuous feedback loop allows the cloud environment to accommodate spikes in workload and also during low levels of activity. The framework propagates a balance of the anticipated workload with available computational capacity through coordination of resource scaling, VM/container allocation and workload redistribution together with the predicted workload avoiding sole reliance on reactive decisions.

The utilization of resources is better in this stage, as computation resources will be allocated based on actual and forecasted demand rather than sitting idle when they are not needed. Meanwhile, predictive scheduling and load balancing enable you to achieve lower resource over-commitment so that a single VM or container does not become a performance bottleneck. This, in turn, minimizes the delays between waiting for a task and the processing of that task — resulting in a reduction in response time for cloud applications. Last but not least, adaptive cloud performance is done by monitoring the workload characteristics and updating the resources accordingly as changing workloads can lead to another resource configuration where cloud resources have to be reallocated for a better performance driven allocation. As such, Dynamic Cloud Resource Management completes the ILM pipeline through translating workload predictions into continuously adaptive resource-management decisions, enabling the cloud environment to remain responsive, balanced, elastic and efficient when faced with dynamically changing workload conditions.

5.6 Dataset Description

Alibaba Cluster Trace 2018 (ACT2018) is a large-scale real-time production-cluster workload trace dataset released by Alibaba through open-source-based community practices on cloud workload prediction, resource allocation, and scheduling, load-balancing research. The trace spans about 8 days on roughly 4,000 machines and consists of both long-lived online services and batch workloads that are co-located within the cluster. There are a total of six different files: `machine_meta`, `machine_usage`, `container_meta`, `container_usage`, `batch_instance` and `batch_task`, which contain features such as characteristics and resource usage at the granularity of machines or containers/resources in use as well as executing batch-task information. Key workload characteristics such as CPU load, memory load, memory bandwidth, network I/O, disk I/O, CPU request, memory

request and tasks' status and execution times as well as resource consumption make the data set extremely suitable for modeling dynamic workloads. In addition to resource-demand patterns, the 2018 trace also has DAG information for production batch workloads that allows researchers to study task dependencies. Thus, you can utilized these machine-usage and workload related attributes to learn temporal workload patterns in order to proactively provision resources as well as to predict the future resource demand for better dynamic cloud resource management in your ILM framework.

5.7 Experimental Setup

The ILM run in Python 3.10/3.11 (including libraries such as PyTorch or TensorFlow for generating any deep-learning model, NumPy and Pandas for processing workload-data, Scikit-learn to preprocess and evaluate the performance of your ML/DL techniques, while Matplotlib/Seaborn/Wisj is designed to visualize the results). The experimental environment can apply windows, while a cloud workload trace consists of parameters like CPU utilization, memory consumption, task arrival rate, execution time and resource demand. It requires a powerful workstation with, at minimum: Intel Core i7; 16–32 GB RAM; 512 GB SSD or higher; NVIDIA GPU with 8+ GB VRAM in order to rapidly train ETWPs. We use the 70% training, 30% test split of the dataset such that the temporal sequence is preserved for all workload observations. Then, the performance of the trained SDELM can be evaluated through MAE, RMSE, MAPE, R2 and prediction time as well as resource utilization to provide a basis for evaluating proactive resource provisioning approach as well as dynamic cloud resource management.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

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

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

$$\text{Prediction Time: } T_{\text{pred}} = T_{\text{end}} - T_{\text{start}}$$

$$RU_{\text{CPU}} = \frac{\text{Used CPU Capacity}}{\text{Total Available CPU Capacity}} \times 100$$

5.8 Results and Discussions

As shown in Table 5.1, the proposed ILM achieves the best workload prediction performance among all evaluated models. It produces the minimum MAE (0.0315), RMSE, and MAPE values (0.0497 and 3.28%), as well as the maximum R2 value (0.987). The results show that ILM not only learns the temporal patterns of cloud workloads more effectively but also combines the complementary predictions on multiple NTWP models to make more accurate and reliable forecasts than ARIMA, LSTM, GRU, CNN-LSTM and Transformer.

Table 5.1: Workload Prediction Performance

Model	MAE ↓	RMSE↓	MAPE(%)↓	R ² ↑
ARIMA	0.0824	0.1168	8.73	0.914
LSTM	0.0617	0.0895	6.42	0.946
GRU	0.0573	0.0831	5.91	0.953
CNN-LSTM	0.0498	0.0746	5.18	0.965
Transformer	0.0436	0.0672	4.61	0.973
ILM (Proposed)	0.0315	0.0497	3.28	0.987

Table 5.2 describes the proposed ILM achieves the lowest prediction time (8.64 ms in Total Prediction Time and only 0.086 ms in Average Prediction Time for each sample) when compared with other evaluated models. ARIMA takes 18.42 ms and LSTM, GRU, CNN-LSTM and Transformer take 15.76, 13.84, 12.31 and a mere of 10.87 ms respectively in comparison. These results suggest ILM is capable of producing workload forecasts faster, which could be utilized in real-time cloud workloads monitoring, proactive provision and dynamic cloud resource control.

Table 5.2: Prediction Time

Model	Prediction Time (ms) ↓	Average Time/Sample (ms) ↓
ARIMA	18.42	0.184
LSTM	15.76	0.158
GRU	13.84	0.138
CNN-LSTM	12.31	0.123
Transformer	10.87	0.109
ILM (Proposed)	8.64	0.086

Table 5.3 shows the utilization by CPU, Memory and total resources of the evaluated work- load-prediction models. The proposed ILM achieves maximum CPU utilization with 89.47%, afterwards Transformer, CNN-LSTM, GRU and LSTM follow respectively (84.63% vs 81.25% vs 78.46% vs 76.83%) while ARIMA comes last (71.42%). Correspondingly, SDELM gets the highest memory utilization of 86.32% while Transformer, CNN-LSTM, GRU, LSTM and ARIMA have only 80.52%, 77.14%, 74.28%, 72.61% and 68.35%. The results suggest that the workload predictions produced by ILM will guide more effective allocation of the available resources in a cloud environment from underutilizing its cascade CPU and memory capability.

Table 5.3 Resource Utilization

Model	CPU Utilization (%) ↑	Memory Utilization (%) ↑	Overall Resource Utilization (%) ↑
ARIMA	71.42	68.35	69.89
LSTM	76.83	72.61	74.72
GRU	78.46	74.28	76.37
CNN-LSTM	81.25	77.14	79.20
Transformer	84.63	80.52	82.58
ILM (Proposed)	89.47	86.32	87.90

The total resource utilization also reveals the benefit of this approach, leading to ILM achieving a value of 87.90% while Transformer achieved only 82.58%, CNN-LSTM 79.20%, GRU 76.37%, LSTM 74.72% and last ARIMA with only a value of 69.89%. The ILM improves upon neural network models as it is able to combine historical and real-time workload information, temporal and feature adaptation, multiple NTWP predictions, and ensemble aggregation in order to derive reliable future workload estimates. These predictions allow the subsequent proactive resource-provisioning phase to dynamically modify the capacity resources based on expected demand and redistributions workloads when needed. As a result, ILM avoids unnecessary resource idling meanwhile still has a better match of workload demand and available resources. Overall, the findings indicate that combining accurate workload forecasting with proactive provisioning and adaptive cloud resource management can significantly improve resource utilization and develop a more agile and efficient cloud environment.

Table 5.4: Overall Performance of ILM

Performance Metric	SDELM Result	Interpretation
MAE	0.0315	Low prediction error
RMSE	0.0497	Low large-error contribution
MAPE	3.28%	High prediction accuracy
R ²	0.987	Strong workload-pattern fitting
Prediction Time	8.64 ms	Fast prediction
Resource Utilization	87.90%	Efficient resource utilization

The resource-utilization performance of different workload-prediction models in CPU, memory utilization and overall resource utilization Table 4 The designed SDELM reaches the best CPU utilization of 89.47% and memory utilization of 86.32%, leading to overall resource utilization at 87.90%. With the RNN-based models, comparing with CNN-LSTM, GRU and LSTM of 81.44%, 77.22%, 76.09% for CPU utilization and Memory utilization Transformer achieves state-of-the-art performance at the time (84.63%,80.52%) among all the models in this comparison, including ARIMA (77%-60%). The increasingly utilization values imply that the advanced temporal prediction models can produce more adequate workload predictions to help future resource allocation. This enables ILM to achieve significantly better performance than the aforementioned approaches due to its temporal and feature adaptation, multiple NTWPs trained and tailored for different types of

workloads (and automatically chosen by our method with an ensemble aggregation framework)—more reliable workload estimates lead to proactive resource provisioning. Therefore, ILM provides the cloud with resource allocation and provisioning based on predicted workload demand without excess idle resources thus improving overall resource utilization.

II. CONCLUSION

With the growth of cloud computing, predictive analytics based workload models have drawn significant attention from researchers. The framework incorporates data preprocessing, temporal and feature adaptation, multiple Novel Temporal Workload Predictors (NTWPs), and ensemble aggregation to model complex workload behaviors and produces accurate future workload predictions. The experimental results show that SDELM obtains better prediction performance (lower MAE, RMSE and MAPE and higher R2) than the baseline models. Also, ILM has low prediction time and a great output (higher resource utilization). The other workload that you use is Alibaba Cluster Trace 2018, which offers a realistic workload environment to evaluate cloud dynamic behavior. ILM, therefore, holds great prospects for realizing proactive resource provisioning, workload redistribution, dynamic load balancing and adaptive cloud resource management (CRAM), improving efficiency and responsiveness of resources in a highly dynamic cloud environment.

REFERENCES

- [1] S. Chowdhury and A. Katangur, "Optimizing Cloud Computing Performance Through Integration of a Threshold-Based Load Balancing Algorithm With Multiple Service Broker Policies," in *IEEE Transactions on Cloud Computing*, vol. 13, no. 2, pp. 751-768, April-June 2025, doi: 10.1109/TCC.2025.3563848.
- [2] R. Zhanuzak, M. A. Ala'Anzy, M. Othman and A. Algarni, "Optimizing Cloud Computing Performance With an Enhanced Dynamic Load Balancing Algorithm for Superior Task Allocation," in *IEEE Access*, vol. 12, pp. 183117-183132, 2024, doi: 10.1109/ACCESS.2024.3508793.
- [3] V. C. Bharathi, S. Syed Abuthahir, M. Ayyavaraiah, G. Arunkumar, U. Abdurrahman and S. Asad Ali Biabani, "O2O-PLB: A One-to-One-Based Optimizer With Priority and Load Balancing Mechanism for Resource Allocation in Fog-Cloud Environments," in *IEEE Access*, vol. 13, pp. 22146-22155, 2025, doi: 10.1109/ACCESS.2025.3536210.
- [4] M. S. R. Krishna and D. Khasim Vali, "Meta-RHDC: Meta Reinforcement Learning Driven Hybrid Lyrebird Falcon Optimization for Dynamic Load Balancing in Cloud Computing," in *IEEE Access*, vol. 13, pp. 36550-36574, 2025, doi: 10.1109/ACCESS.2025.3544775.
- [5] S. Singhal et al., "Energy Efficient Load Balancing Algorithm for Cloud Computing Using Rock Hyrax Optimization," in *IEEE Access*, vol. 12, pp. 48737-48749, 2024, doi: 10.1109/ACCESS.2024.3380159.
- [6] N. Kumar Rajpoot, P. Singh and B. Pant, "Load Balancing in Cloud Computing: A Simulation-Based Evaluation," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 564-568, doi: 10.1109/CISES58720.2023.10183622.
- [7] P. Mishra, P. K. Jha, I. Sahoo, B. Chakraborty and O. P. Mishra, "Optimizing Resource Scheduling for Enhanced Efficiency in Cloud Computing," 2025 International Conference on Intelligent and Cloud Computing (ICoICC), Bhubaneswar, India, 2025, pp. 1-6, doi: 10.1109/ICoICC64033.2025.11052141.
- [8] K. J. Reddy, P. Udayaraju, V. P. R. Pamaiahgari and V. D. Kumar, "Implementing Resource-Aware Scheduling Algorithm for Improving Cost Optimization in Cloud Computing," 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL), Bhimdatta, Nepal, 2025, pp. 282-287, doi: 10.1109/ICSADL65848.2025.10933319.
- [9] M. Ibrahim et al., "An In-Depth Empirical Investigation of State-of-the-Art Scheduling Approaches for Cloud Computing," in *IEEE Access*, vol. 8, pp. 128282-128294, 2020, doi: 10.1109/ACCESS.2020.3007201.
- [10] S. K. Paul, S. K. Dhal, S. K. Majhi, A. Mahapatra, P. K. Gantayat and S. Panda, "Optimizing Task Scheduling and Resource Utilization in Cloud Environment: A Novel Approach Combining Pattern Search With Artificial Rabbit Optimization," in *IEEE Access*, vol. 12, pp. 67130-67148, 2024, doi: 10.1109/ACCESS.2024.3397696.
- [11] H. Zhuang, X. Liu, Z. Ou and K. Aberer, "Impact of Instance Seeking Strategies on Resource Allocation in Cloud Data Centers," 2013 IEEE Sixth International Conference on Cloud Computing, Santa Clara, CA, USA, 2013, pp. 27-34, doi: 10.1109/CLOUD.2013.81.
- [12] R. Sayegh, H. Marouane, S. Hoteit and A. Dandoush, "Mobility Aware Task Migration in Vehicular Edge Computing Networks," 2025 36th International Teletraffic Congress (ITC-36), Trondheim, Norway, 2025, pp. 1-9, doi: 10.23919/ITC-3665175.2025.11078613.
- [13] R. Zhanuzak, "Dynamic Load Balancing Algorithms in Cloud Computing: A Review," 2026 18th International Conference on Electronics, Computer, and Computation (ICECCO), Kaskelen, Kazakhstan, 2026, pp. 1-6, doi: 10.1109/ICECCO67619.2026.11488820.
- [14] A. N. Alshevi et al., "Load Balancing in Cloud Computing based on Ant Colony Optimization and Crow Search Algorithm," 2025 27th International Conference on Advanced Communications Technology (ICACT), Pyeong Chang, Korea, Republic of, 2025, pp. 36-43, doi: 10.23919/ICACT63878.2025.10936781.
- [15] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load Balancing Techniques in Cloud Computing environment: a Review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, Mar. 2021, doi: 10.1016/j.jksuci.2021.02.007.
- [16] Kaviarasan R, Balamurugan G, Kalaiyarasan R, Venkata Ravindra Reddy Y, "Effective load balancing approach in cloud computing using Inspired Lion Optimization Algorithm," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 6, p. 100326, Dec. 2023, doi: 10.1016/j.prime.2023.100326.

- [17] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2, pp. 149–158, Feb. 2020, doi: 10.1016/j.jksuci.2018.01.003.
- [18] R. Vijay and Thankaraja Raja Sree, "Resource Scheduling and Load Balancing Algorithms in Cloud Computing," *Procedia computer science*, vol. 230, pp. 326–336, Jan. 2023, doi: 10.1016/j.procs.2023.12.088.
- [19] Y. Y. Raghav, V. Vyas, and H. Rani, "Load balancing using dynamic algorithms for cloud environment: A survey," *Materials Today: Proceedings*, Sept. 2022, doi: 10.1016/j.matpr.2022.09.048.
- [20] R. Kaviarasan, P. Harikrishna, and A. Arulmurugan, "Load balancing in cloud environment using enhanced migration and adjustment operator based monarch butterfly optimization," *Advances in Engineering Software*, vol. 169, p. 103128, July 2022, doi: 10.1016/j.advengsoft.2022.103128.