Computing and Informatics, Vol. 43, 2024, 64–93, doi: 10.31577/cai_2024_1_64

ETSA-LP: ENSEMBLE TIME-SERIES APPROACH FOR LOAD PREDICTION IN CLOUD

Shveta Verma, Anju Bala

Department of Computer Science
Thapar Institite of Engineering and Technology
Patiala, Punjab, India
e-mail: {shveta.verma, anjubala}@thapar.edu

Abstract. Cloud Computing has immersed researchers in accessing the resources on-demand for deploying various applications by offering infinite services. But, as the demand for cloud resources is dynamic, it significantly affects the load on the system. Thus, this research emphasizes deploying a dynamic and autonomic load prediction framework. This paper proposes an Ensemble Time-Series Approach for Load Prediction (ETSA-LP), which integrates various time-series analysis techniques for predicting CPU and memory utilization. To evaluate the efficiency of the proposed approach, a series of experiments on Google and PlanetLab traces have been conducted in a real Cloud environment. The results were compared according to different performance metrics and models, the accuracy determined and the minimal error rate selected as the best among others. The proposed ensemble approach gives the best performance over the existing models showing the remarkable accuracy improvement and reducing the error rate and execution time.

Keywords: Cloud computing, load prediction, time-series forecasting, ensemble approach, dynamic resource prediction

1 INTRODUCTION

Cloud Computing refers to a computing network model that supports single and multiple programs or applications to be executed on an inter-connected set of servers rather than on a native machine. For vast data centers of Cloud Computing, it is required for physical and virtual interrelated networks to communicate with faster

Giga-bit speed. But, this initial requirement gives rise to the problem of predicting the resource availability in the system for upcoming processes in advance. As the present computer systems and their workloads are dynamic, it becomes challenging to make such autonomous predictions. Cloud Computing platforms must be able to propose and acquire resources quickly to certify high scalability, flexibility, and effective cost [1]. This will also help sustain the prerequisites of various Cloud-based applications by dynamic load prediction for appropriate resource utilization.

Based on a Cloud system's estimated future load and performance, users can make desired decisions to prevent the system from traffic flow caused by peak load. There should be an accurate simulation of the correlation between historical and future values for precise and intelligent load prediction in Cloud Computing systems. Proper knowledge of backend workloads is also necessary [2]. Diverse models have been proposed till now for load prediction based on the history of jobs executed or interrelated resources. In addition, many models' time-varying resources and non-adaptive features affect the performance of cloud data centers [3]. Hence, newly refined algorithms such as time-series analysis techniques to predict load from multiple hosts proactively. The reasons why autonomic load prediction in Cloud is recommended are:

- The first reason is that most recent applications in the Cloud (e.g., e-health, smart homes, smart cities, etc.) have varying loads because of their resource consumption, power usage and configuration details that change from time to time. This results in complex and unpredictable behavior of resources and has a major effect on a load of a Cloud System.
- Due to the ever-increasing demand for resource usage and intrusion between the applications accommodated on physical machines, it becomes crucial to design and implement autonomic resource usage prediction in terms of load [1].
- When allocating several requests to a single VM, it may result in the over-provisioning of resources. Due to the reason that number of requested resources becomes more than the number of available resources, the performance features of an application assigned to the VM will degrade. Hence, a prediction model must be able to notify about the under-provisioning or over-provisioning situation of resources prior to the resource manager [4].
- As there are high inconsistencies in cloud performance metrics, the existing load prediction models usually fail to offer appropriate accuracy. Therefore, an intelligent ensembling technique combining prediction results from different models can proficiently schedule cloud resources in this scenario.

The key focus of the proposed research work is to accomplish an autonomic ensemble load prediction technique by exploring and comparing the prediction outcomes of time-series approaches. The five time-series approaches are used as base predictors for the ensemble, namely, Auto-Regressive Integrated Moving Averages (ARIMA), Artificial Neural Networks (ANN), Support Vector Machines (SVM), Long Short-

Term Memory (LSTM) and Exponential Smoothing (ES). These models have been verified to carry out a prediction-based ensemble model for identifying over-loaded as well as under-loaded hosts. The models with determined accuracy and most minor error have been selected as the best among others and by deploying on Google and PlanetLab traces on the Google Cloud platform.

1.1 Motivations and Contributions

- As the data size increases daily in Cloud, there is a need to focus on predicting and managing the load on the system. For precise and intelligent load prediction, there should be an accurate simulation of the correlation between historical and future values [2]. In this paper, the problem of load prediction using CPU and memory utilization is analyzed so that Cloud systems could handle under-provisioning or over-provisioning situations of resources intelligently.
- There are several load prediction techniques used by numerous authors, such as queuing theory, reinforcement learning, etc. Still, all these static techniques cannot extrapolate patterns and encapsulate the time-based components. Hence, there is a requisite to deploy time-series analysis techniques for efficient and autonomic load prediction [5]. Modeling a framework for load prediction using five time-series analysis techniques (ARIMA, ANN, SVM, LSTM and ES) and estimating the CPU and memory usage more precisely has been done in this paper.
- Though much research has been done on load prediction and time-series models, an inclusive methodology must be used to combine different prediction models. More profoundly, one prediction model would effectively predict some trends, but the same would be imprecise in other trends due to its dynamic performance. This challenges the ensemble models to consider such techniques, which results in the best accuracy and productivity metrics [6]. Therefore, this research work proposes an Ensemble Time-Series Approach for Load Prediction (ETSA-LP) model using five base prediction models.
- Based on the objective of load prediction in Cloud applications, multiple QoS metrics needs to be explored and validated to increase the system's performance. The essential QoS parameters, such as response time, CPU usage, memory usage, prediction accuracy, etc., need to be analyzed, which can directly or indirectly affect an application's performance [7]. Execution of experiments has been performed to assess the performance of the recommended ETSA-LP algorithm with the traces of Google and PlanetLab datasets regarding the accuracy and error benchmarks.

1.2 Paper Organization

The rest of the paper is organized as follows: Section 2 represents related work. Section 3 briefly introduces preliminaries and evaluation metrics for validating the

proposed approach. Section 4 describes the proposed load prediction framework ETSA-LP and the algorithm used for ensembling. The dataset description, followed by experimental results and analysis, is discussed in Section 5. Finally, Section 6 highlights future work and conclusion.

2 RELATED WORK

As we discussed above, recent growth in autonomic resource management has led to increased concern in the proactive load prediction. Numerous techniques have been recommended for load prediction using several innovative methods, metrics and applications. In this section, a survey of the work related to ensembling approaches for predicting load has been conducted.

Although many prior research works were performed using time-series forecasting for load prediction, the foremost challenge of choosing a suitable and accurate predictive model still needs to be investigated. One of the solutions to choosing a wise model is to assess the existing predictions from previous forecasting models with current models. For instance, some authors proposed specific recurrent machine-learning methods such as Long Short-Term Memory (LSTM) [5, 8]. Also, LSTM models can give better results as compared to ARIMA, which are frequently reliant on the self-made features demanding expert knowledge in the respective field [9, 10, 11]. On the other hand, many non-linear models (e.g. ANN) were suggested in the literature to overcome the challenge of linear statistical time-series models [12, 13].

Cao et al. [2] proposed a novel dynamic ensemble model which consisted of two layers – prediction optimization and an ensemble layer. The proposed approach improved prediction accuracy by 4.81 % and MREs have been enhanced by 7.37 %. Belouch and Hadaj [14] compared bagging, boosting and stacking ensemble techniques to increase the performance of intrusion detection systems. The base prediction models used were decision tree (J48), Naïve Bayes (NB), Multi-Layer Perception (MLP) and REPTree.

Chen and Wang [9] proposed a resource demand prediction method named EEMD-ARIMA (Ensemble Empirical Model Decomposition-ARIMA) to accurately predict the number of VMs and CPU cores used. Rahmanian et al. [15] proposed an ensemble method based on Learning Automata (LA), combining the prediction values of all base models according to their weights and performance. Similarly, Ghobaei-Arani et al. [16] developed an algorithm using LA theory, correlation coefficient and ensemble method for better VM placement, less energy consumption and SLA violations.

Kaur et al. [4] proposed an intelligent Regressive Ensemble Approach for Prediction (REAP) using feature selection with Genetic Algorithm (GA) to predict resource usage. Compared with existing models, the proposed approach improved accuracy by 2 % and reduced execution time by 16.2 %. Nguyen et al. [10] developed an ensemble model for workload time-series prediction named as ESNemble

(Echo State Network) which is based on Neural Networks (NN). Mauldin et al. [17] performed an ensemble of deep learning techniques RNN (recurrent NN) along with stacking and AdaBoost for IoT applications. Toumi et al. [18] proposed an RTSLPS (Real Time Server Load Prediction System) technique based on incoming task classification and VM interference detection which used ensemble drift detectors.

Shaw et al. [12] proposed PIEA (Predictive Interference and Energy Aware) ensemble approach to improve energy efficiency and performance for dynamic VM placement consolidation. Kumar et al. [6] presented an ensemble workload forecasting method using ELM (Extreme Learning Machine) and also proposed a new metric RelMAE (Relative MAE) which resulted in 99.20 % improvement. Tuli et al. [19] proposed a novel framework HealthFog for automatic heart disease detection and analysis. The ensemble deep learning NN and PCA have been used for data feature extraction and reduction, resulting in improved prediction accuracy and Quality of Service (QoS).

Some authors also proposed workload and resource estimation techniques other than time-series analysis such as reinforcement learning (RL), queuing theory, threshold-based, etc. In the case of threshold-based approaches, the setting of the threshold value is a time-consuming process and the effectiveness of the rules under burst workloads still need to be determined. Conversely, the main drawback of queuing theory models is that they must be re-analyzed each time if there are changes in an application or the workload [7]. Most of the RL approaches refer to having ample state space means the number of states frequently increases with the number of state variables, resulting in scalability problems [20]. Besides all these approaches, time-series analysis techniques have been widely used to provide the best results and optimal performance in dynamic load prediction for the Cloud environment.

Table 1 represents the existing works on ensemble load prediction techniques underlining their quality metrics and outcomes. Table 2 highlights these current works by comparing the type of technique, predicted resource (CPU or memory), dataset (single or multiple) and performance metrics with the proposed approach.

Though much research has been done on load prediction and ensembled time-series models, it still needs to be solved. Most of the proposed ensemble algorithms were implemented using the combination of linear and non-linear prediction models [15]. An inclusive methodology to combine different types of prediction models is still needed. More profoundly, one prediction model would be effective in predicting some trends but the same would be imprecise in other trends, due to its dynamic performance. Hence, most state-of-the-art ensemble approaches are productive in a particular constituent model [27].

In this paper, an initiative has been taken to propose a more precise ensemble prediction model ETSA-LP that integrates the prediction values of its primary models by giving weight to each prediction. Specifically, this research compares statistical, neural, and ensemble machine-learning models and proposes an ensemble framework that combines the best forecasts of base models. The working of the proposed ETSA-LP technique for predicting CPU and memory load on differ-

Year	Proposed Technique	Prediction Method	Quality Metrics	Dataset and Simulation	Outcome
2017 [14]	Comparison of ensemble models	Decision Tree (J48), NB, NN and REP-Tree	Accuracy, False Positive rate, ET	UNSW-NB15 dataset	High accuracy and fast execution time
2018 [9]	EEMD-ARIMA	ARIMA	RMSE, MAE, MAPE	Random number of VMs and CPU chores	Outperforms ARIMA in short-term predictions
2018 [11]	Ensemble model using BIC and Smooth filters	ETS, SES, Holt's Winter, DES and Automatic ARIMA	MAE, RMSE, MAPE, MASE, PRED	PlanetLab traces	Outperforms GA and other models
2018 [15]	LA based ensemble algorithm	LA	RMSD and error ratio	PlanetLab traces	Achieved better results
2018 [16]	LA and ensemble algorithm for VM placement	LA	MCC, average number of migrations	CloudSim	Reduced energy consumption and SLA violation
2018 [4]	REAP	GA, BRR, NN, SVM, DT, ELM, RF, LM, BRNN	Correlation coefficient, RMSE, ET, accuracy	Cybershake workflow	Improved accuracy by 2 % and reduced ET by 16.2 %
2019 [10]	ESNemble	GA, NN, NB, AR, ARIMA, ARMA, ETS	response time, MAE, MAPE, RMSE	CRAN, EDGAR and Kyoto datasets	Outperforms existing algorithms
2019 [17]	Ensemble Deep learning techniques	Recurrent NN, AdaBoost	Recall, Precision, Specificity, Accuracy	SmartFall and UniMiB SHAR Dataset	Outperforms single RNN

to be continued

Year	Proposed Technique	Prediction Method	Quality Metrics	Dataset and Simulation	Outcome
2019 [18]	RTSLPS	Adaptive random Forest, OzaBagADWIN	Classification Accuracy, KAPPA statistics, ET	Amazon EC2 traces and Google CE datacenter	Results in 96.5 % accuracy
2020 [8]	Wavelet-GMDH-ELM (WGE)	GMDH, ELM, LSTM, RMSE, MAPE, MSE, MAE	Worldcup, Intel Netbatch logs	Internet traffic data of TSDL	Improved 8 % MAPE
2020 [5]	Ensemble using exponential weighting	ARIMA, MLP, LSTM	RMSE, R-Square	Truven MarketScan dataset	Improved RMSE and R-Square
2020 [21]	RSA-NFTSAR	Hybrid NN, ANN	MSE, MAPE, RMSE	Matlab r2015, PlanetLab traces	Improved 33.83 % MAPE, 36.54 % MAE and 36.70 % RMSE
2020 [12]	PIEA algorithm	LR, ANN, SVM	Overall accuracy, precision, recall, energy consumption, SLA violations	Real traces of Microsoft Azure	Improved energy efficiency by 34 % and reduced SLA by 77 %
2020 [6]	Expert learning using ELM	ELM, NN	MSE and RelMAE	Google and PlanetLab traces	Improved 99.20 % RMSE
2020 [19]	healthFog framework using ensemble deep learning	PCA, Deep NN	Power consumption, latency, ET, n/w bandwidth	Cleveland dataset	Better QoS and prediction accuracy
2021 [22]	BG-LSTM model	LSTM, Grid LSTM	RMSLE, MSE	Google traces	Outperforms traditional models

to be continued

Year	Proposed Technique	Prediction Method	Quality Metrics	Dataset and Simulation	Outcome
2021 [23]	Ensemble approach	ANN, SVR, M5Rules	RMSE, MAE, MAPE	Benchmarking Data Source	Improved 123.4 % MAE and 209.3 % MAPE
2021 [24]	CC resource predictor	SVM, DT, k-NN, FNN	Accuracy, RMSE, FI-score	Phold and SOS datasets	higher accuracy by 4 %-20 %
2021 [25]	Cocktail framework	Pre-trained models for image classification	Accuracy, latency	ImageNet dataset, AWS EC2	Reduced latency by 2 % and achieved 96 % accuracy
2021 [26]	E2LG algorithm and GAN-LSTM	Stacked LSTM	Accuracy, RMSRE, MAPE, MeAPE	3 real Cloud workload traces	Improves accuracy 5 %-12 %
Prop.	**ETSA-LP**	**ARIMA, SVM, ANN, LSTM, ES**	**Accuracy, RMSE, ET**	**Google traces and PlanetLab**	**Improved accuracy, error and execution time**

Table 1. Review of existing ensemble-based load prediction mechanisms

Ref.	TS	EB	CPU	MEM	SD	MD	ACC	RMS	ET
[14]	✓	✓	✗	✗	✓	✗	✓	✗	✓
[9]	✓	✓	✓	✗	✓	✗	✓	✓	✗
[11]	✗	✓	✓	✗	✓	✗	✓	✓	✗
[15]	✗	✓	✓	✗	✓	✗	✗	✓	✗
[4]	✓	✓	✓	✗	✓	✗	✓	✗	✓
[16]	✗	✓	✓	✗	✓	✗	✗	✗	✗
[10]	✓	✓	✓	✗	✗	✓	✓	✗	✓
[17]	✗	✓	✗	✗	✗	✓	✓	✗	✗
[18]	✓	✗	✓	✓	✗	✓	✓	✗	✓
[8]	✓	✓	✗	✗	✗	✓	✓	✓	✗
[5]	✓	✓	✗	✗	✓	✗	✗	✓	✗
[21]	✓	✓	✓	✗	✓	✗	✓	✓	✗
[12]	✓	✓	✓	✗	✓	✗	✓	✗	✗
[6]	✗	✓	✓	✓	✗	✓	✓	✓	✗
[19]	✗	✓	✗	✗	✓	✗	✓	✗	✗
[22]	✗	✗	✓	✓	✓	✗	✓	✗	✗
[23]	✓	✓	✗	✗	✓	✗	✓	✓	✗
[24]	✓	✓	✓	✗	✗	✓	✓	✗	✗
[25]	✓	✓	✓	✗	✗	✓	✓	✗	✓
[26]	✗	✓	✓	✗	✗	✓	✓	✓	✓
PA	✓	✓	✓	✓	✓	✓	✓	✓	✓

TS: Time-Series techniques; *EB*: Ensembling; *CPU*: CPU load, *MEM*: Memory load, *SD*: Single Dataset; *MD*: Multiple Datasets; *ACC*: Accuracy; *RMS*: Root Mean Square Error; *ET*: Execution Time; *PA*: Proposed Approach

Table 2. Summary of the existing related work

ent datasets has been explored. Furthermore, the experimental results have been presented, including comparing the proposed ensemble model predictions with the existing models.

3 PRELIMINARIES

3.1 Load Prediction in Cloud

With the advancement of private and public cloud data centers, leasing virtual machines to host applications is quite common. Usually, Cloud tenants do not choose to pay for the resources they attain but are not consumed when the load is light. This concern would be feasible only if the upcoming load can be predicted earlier before validating Quality of Service (QoS) parameters [28]. Furthermore, there can be a probability of performance degradation in case of heavy load [29].

Generally, Amazon EC2 service providers recommend resources on a VM basis and allow VMs to be added, released, or migrated according to the load divergence. So, it is enormously endorsed for cloud service providers to offer finer-grained auto-

nomic facilities that dynamically attain resources according to an application request and permit cost based on the VM size [30]. Also, there should be an accurate simulation of the correlation between historical and future values for precise and intelligent load prediction in Cloud Computing systems. Moreover, proper knowledge of back-end workloads is necessary. Therefore, predicting over-utilized and under-utilized hosts requires more sophisticated algorithms such as time-series analysis techniques to predict the load proactively.

3.2 Time-Series Analysis

A time-series method organizes data points to be measured at constant time intervals. This technique can be applied to regulate self-repetitive input workload patterns or foresee forthcoming values [7]. For instance, every performance parameter such as average CPU load will be calculated at fixed intervals. Therefore, time-series analysis is the most prominent resource usage and workload prediction approach in Cloud Computing.

Efficient forecasting works with clean, time-stamped data and can identify genuine trends and patterns in historical data [31]. But, this contrasts while working on static data. Also, the choice of algorithms in time-series techniques differs entirely from those in static. Most static machine learning algorithms like linear regression do not have this capability as they generalize the training space for any new prediction. In this context, machine learning-based models (such as SVM, LSTM, ANN, ES) are attaining inclusive attention in forecasting load nowadays because of their introspection aspect for multiple applications [32]. So, the primary focus of this paper is to evaluate different statistical, neural, and ensemble techniques in their ability to predict resource load. This paper uses the following five models as base predictors:

- Auto-Regressive Integrated Moving Averages (ARIMA): This model defines the prospect value of a variable that is supposed to be a linear function of the last few observations and some random errors. This statistical model is popular and could be a good baseline for comparing other models because of its simplicity [5]. However, one challenge in the ARIMA model is its pre-assumption of linearity in the underlying time series, which may need to be improved in various practical scenarios that contain non-linear time-series data.
- Artificial Neural Networks (ANN): ANN technique can solve many problems in classification and long-term forecasting compared to linear statistical models. This model belongs to the data-driven approach, where training depends on the available data with little prior rationalization regarding relationships between variables. ANNs are self-adaptive as they do not make assumptions about the underlying time series' statistical distributions and can naturally perform non-linear modeling [28].
- Support Vector Machines (SVM): This model can categorize linear data accurately and is also used for complex decision boundaries because of its lower

complexity. This classic method is generally used in classification, regression, etc. to solve complex real-world problems with data having multiple input features.

- Long Short-Term Memory (LSTM): A type of neural network with the capability to memorize the past values in the network is known as the LSTM model. LSTM is a supervised deep learning method widely used in time series nowadays [26].
- Exponential Smoothing (ES): Exponential smoothing is an alternative linear model dependent on fundamental historical observation values to make predictions. It weighs so that recent observations are much heavier than old observations.

3.3 Ensembling Approach

Though much research has been done on load prediction and time-series models, there is a need for an inclusive methodology to combine different types of prediction models. More profoundly, one prediction model would effectively predict some trends, but the same would be imprecise in other trends due to its dynamic performance. In conventional ensembling techniques, the most often occurred, or the average prediction from multiple models is chosen as the concluding prediction outcome from the ensemble [12]

This research proposes a dynamic Ensemble Time-Series Approach for Load Prediction (ETSA-LP) that combines the best prediction results from compound models (ARIMA, ANN, SVM, LSTM and ES) to predict CPU and memory utilization. A dynamic exponential weighting algorithm has been proposed, in which a distinct model's best prediction outcome selected from diverse models has been dynamically weighted. A brief discussion of this proposed algorithm is given in the upcoming sections.

3.4 Evaluation Metrics

The QoS attributes/metrics, such as response time, CPU load, throughput, accuracy, cost, etc., are usually part of an SLA and are frequently changing. These parameters must be monitored according to user requirements to impose the agreement. Numerous researchers have used different load prediction techniques in Cloud to effectively implement their systems with specific metrics [7]. As there may be many user requests in multiple service queues, existing load prediction systems need more evaluation metrics for dynamic user interactions with the system. Therefore, QoS parameters like CPU usage, memory usage, error rate, response time, and so forth they can be added to test the system's performance. In this paper, the base predictors ARIMA, ANN, SVM, ES and LSTM have been compared with the proposed ETSA-LP to assess the best performance using the following metrics.

3.4.1 RMSE: Root Mean Square Error

For experimental evaluation, RMSE is calculated to measure the error in percentage. RMSE is defined in Equation (1) where X_t is the actual output, x_t is the predicted output and n specifies the total number of observations in the dataset. The lower value of RMSE validates to be a more precise prediction technique.

$$RMSE = \sqrt{\frac{1}{n}} \sum_{t=1}^{n} (x_t - X_t)^2. \tag{1}$$

3.4.2 R (Coefficient of Correlation) and r^2 (Coefficient of Determination)

R is a measure of the goodness-of-fit, which its value falls within the range $[0, 1]$ and is generally applied to the linear regression models [6]. r^2 is a statistical measure to denote how close the data is fitted on the regression line. It is also defined as the coefficient of determination or the coefficient of multiple determination for multiple regression. It is the percentage of the response variable variation that a linear model represents. Specifically, the higher the r^2, the better the model fits the data. The value of r^2 is always between 0 and 100 %:

- 0 % means that the model describes none of the variability of the response data around its mean.
- 100 % means that the model describes all the variability of the response data around its mean.

3.4.3 Accuracy

Accuracy is the degree to which the outcomes of an actual calculation or measurement are grouped around the accurate value. The mathematical notation to compute accuracy is given in Equation (2):

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} * 100. \tag{2}$$

Here, *tp*, *tn*, *fp* and *fn* denote the number of true positives, true negatives, false positives and false negatives, respectively.

3.4.4 Total Execution Time

Total execution time is the overall time a process occupies to finish execution. The formula to calculate total execution time is indicated in Equation (3):

$$TotalTime = t_e - t_s. \tag{3}$$

Here, t_e denotes the finish time for executing data and t_s is the initial time when execution starts actually.

4 PROPOSED ENSEMBLE APPROACH: ETSA-LP

The main components of the proposed ETSA-LP framework are depicted in Figure 1. This consists of a load balancer, load predictor module, resource manager module, data storage module, and Cloud infrastructure in the system. Cloud tenants use different resources or virtual machines to deploy various applications and services on Cloud and these resource utilizations are monitored and profiled by the resource manager. It will also gather efficient metrics to check resources' past and present state. The required metrics (CPU and memory usage) and logs are transferred to the workload database and the historical information. Also, the historical workload information is used to train the prediction model. Load predictor allocates suitable time-series forecasting techniques for the prediction of workload. Moreover, it is responsible for combining the forecasts of base models and applying an ensemble algorithm to find optimal results. The resource manager also dynamically manages the resource allocation according to user request changes. It estimates the number of resources according to the status of the resource monitor. It also dynamically manages the resource allocation according to user request changes and publishes load-balancing instructions to the load predictor. This load-balancing module controls load among various VMs deployed on the Cloud environment.

The methodology used for the proposed load prediction framework is depicted in Figure 2. The first step is to gather logs of different resources Cloud users use as a dataset. Then this dataset was checked and analyzed to determine whether any pre-processing or cleaning was required. The next step is to apply time-series forecasting techniques on the given dataset to compare the accuracy of different models. The time-series forecasting techniques used in this proposed work are ARIMA, NN, SVM, LSTM and ES. The predictions from these base models have been ensembled for efficient and dynamic load prediction using the proposed ETSA-LP. Also, a mathematical formulation is derived for CPU and memory utilization based on different parameters. Further, the effectiveness of the proposed approach is validated by testing on Google Cloud and the results have been compared with existing models. The list of notations used in this approach is presented in Table 3.

4.1 ETSA-LP: Exponential Weighted Algorithm

The working of the exponentially weighted algorithm used for the ensemble in the proposed ETSA-LP is presented in Algorithm 1. The algorithm starts with a given set of N predictions from diverse models on the training data and assigns equal weight to them (line 1). In the second step, the weight of each model is assumed as $1/n$, which verifies that the sum of all model weights is equal to 1 at each step (line 2). Then, for each training sample (e.g. 75 training samples in the case of Google traces), the squared error between each model's prediction and actual data is calculated i.e. RMSE is calculated for different model predictions (line 3). After each training sample, each model's weights are revised using the squared error for other model predictions, where x is assumed as a variable-rate parameter (line 4).

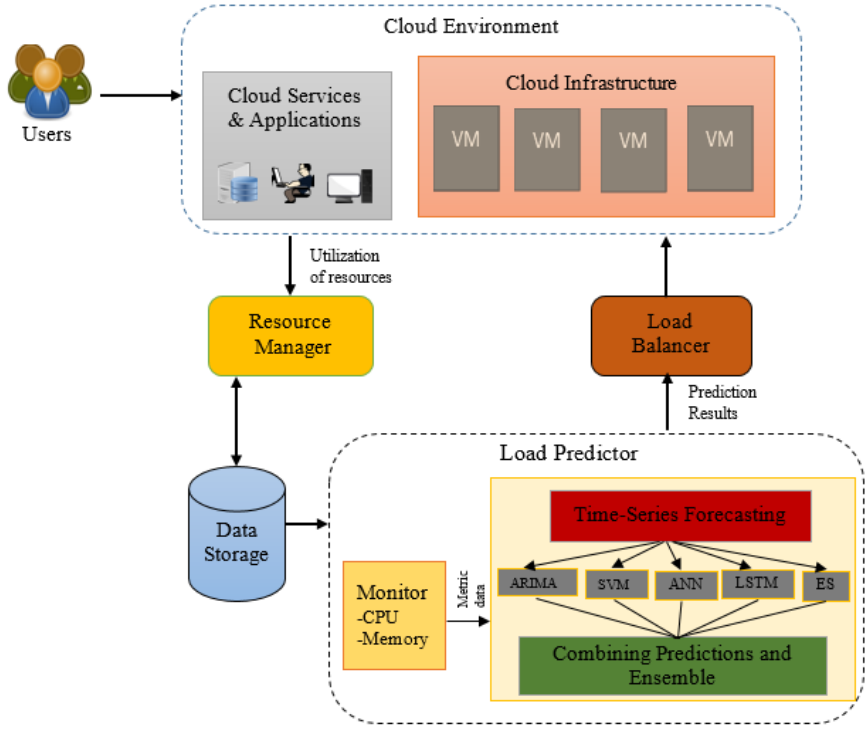


Figure 1. Proposed ensemble load prediction framework: ETSA-LP

Figure 2. Methodology for the proposed ETSA-LP approach

Finally, the weights of each model's predictions are normalized by dividing them by the total sum of the weights across all models (line 5). The normalization of weights is performed to bring all weights between 0 and 1. This process repeats until all training samples are covered. The different values for x are defined to obtain the optimal weight corresponding to each model, for instance, 0.0 to 1.0. These weights are then selected to obtain the minimum value for RMSE and maximum value for R^2.

As discussed above, the different values for the variable-rate parameter x are defined to obtain the optimal weight corresponding to each model. Therefore, 11 dif-

Component	Description
n	Set of predictions
1_n	Weight assigned to each model
T	Training samples
M	Model used
W	Weight of each model (updated)
S	Sum of models weights
x	Variable-rate parameter
P	Prediction outcomes of each model
R	RMSE values
N	No. of nodes
J_N	No. of jobs
V_N	No. of VMs
C_i	Clock cycles per instruction
IC	Instruction count
CR	Clock rate
V_n	No. of VMs running on N^{th} node
$totalCU_N$	Total CPU utilization
$avgCU_N$	Average CPU utilization
$totalMU_N$	Total memory utilization
$avgMU_N$	Average memory utilization

Table 3. Notations used in the article

ferent values of x parameter are tried in this ensemble algorithm. For instance, with $x = 0.3$, the ensemble results were obtained with corresponding weights: ARIMA (0.313), SVM (0.332), ANN (0.329), LSTM (0.300) and ES (0.335). Similarly, when the value of x is taken as 0.8, the ensemble results were obtained with these weights: ARIMA (0.011), SVM (0.231), ANN (0.534), LSTM (0.768) and ES (0.422). The

Algorithm 1 ETSA-LP: Exponential Weighted Algorithm

Require: $M_i, T, P\frac{T}{M_i}$

1. For each model M,
   $W\frac{1}{M_i} \leftarrow \frac{1}{n}, M_i = 1 \ldots nModels$
2. Sum of all models be equal to 1
   $S = \sum_{i=1}^{n}(W\frac{1}{M_i} == 1)$
3. For i=1 to N, Calculate RMSE of models
   $R(P\frac{T}{M_i}Y_t) = (P\frac{T}{M_i} - Y_t)^2$
4. Update weights by adding RMSE
   $W\frac{R}{M_i} + 1 \leftarrow W\frac{R}{M_i} * xR(P\frac{T}{M_i}Y_t)$
5. Normalize the weight of each model
   $W\frac{R}{M_i} + 1 \leftarrow \frac{W\frac{R}{M_i}+1}{\sum_{i=1}^{n} W\frac{1}{M_i}+1}$

overall results and comparison of the proposed ETSA-LP approach based on this ensemble algorithm are discussed in the upcoming sections.

4.2 Dynamic Resource Prediction Algorithm Using CPU and Memory Utilization

The exponential weighted algorithm for ensemble gives the ideal combination of machine learning models and improves the prediction accuracy. This ensemble model is employed on the Google Traces and PlanetLab datasets for prediction. The final output of Algorithm 1 is used as an input parameter in Algorithm 2, which is further used for dynamically predicting CPU and memory utilization. The first step is to initialize the value of the number of jobs and the number of VMs as 1. Then, for each K^{th} job running on the N^{th} node on M^{th} VM, CPU and memory utilization are calculated as shown in step 2. Further, the total CPU utilization is predicted by the formula given below:

$$totalCU_N = \frac{C_i * IC}{CR}. \tag{4}$$

In Equation (4), C_i refers to the clock cycles per instruction, IC is the instruction count and CR denotes the clock rate. By taking this total CPU utilization into account, the average CPU utilization is calculated as follows:

$$avgCU_N = \frac{\sum_{M=1}^{V_N} \sum_{K=1}^{J_N} CU_x}{totalCU_N} * 100. \tag{5}$$

In Equation (5), $avgCU_N$ is calculated at any given time for N^{th} node, considering V_N as the number of VMs running on the N^{th} node and J_N as the number of jobs assigned to V_N VMs. CU_x is the CPU utilization of K jobs running on M VMs on the N^{th} node. Similarly, the average memory utilization is calculated by the formula:

$$avgMU_N = \frac{\sum_{M=1}^{V_N} \sum_{K=1}^{J_N} MU_x}{totalMU_N} * 100. \tag{6}$$

At any given time, for the N^{th} node, the average memory utilization $avgMU_N$ can be given as shown in Equation (6), where V_N is the number of VMs running on the N^{th} node and J_N is the number of jobs assigned to V_N VMs. MU_x is the memory utilization of K jobs running on M VMs on the N^{th} node and $totalMU_N$ is the total memory utilization for the N^{th} node.

5 EXPERIMENTS AND EVALUATIONS

5.1 Dataset Description

Due to great variabilities in Cloud Computing workloads, it is nearly 20 times noisier than Grid Computing workloads. Therefore, for effective validation of our proposed

Algorithm 2 Dynamic Resource Prediction Algorithm (CPU and Memory)

Require: N, M, J, K, C_i, IC, CR

1. Intialize $M = 1$, $J = 1$
2. For each K^{th} job running on N^{th} node on M^{th} VM,
   Calculate CPU and Memory utilization
   $CU_x, MU_x; x = NMK$
3. For each N^{th} node on M^{th} VM,
   Calculate total CPU utilization
   $totalCU_N = \frac{C_i * IC}{CR}$
4. Calculate Average CPU utilization
   $avgCU_N = \frac{\sum_{M=1}^{V_N} \sum_{K=1}^{J_N} CU_x}{totalCU_N} * 100$
5. Calculate Average memory utilization
   $avgMU_N = \frac{\sum_{M=1}^{V_N} \sum_{K=1}^{J_N} MU_x}{totalMU_N} * 100$
6. Return $avgCU_N, avgMU_N$

approach ETSA-LP, two real cloud workload traces from Google and PlanetLab have been used as baseline benchmarks. Google traces includes the data of 29 days collected from Google's cluster cell. This workload is tested every 5 minutes and contains 75 samples inclusive of 9 218 jobs and 176 580 tasks [33]. In the experiments, the resource usage proportion comprises the traces of CPU and memory demands of VM instances hosted on the cluster, as shown in Table 4. All these measurements are regulated between 0 and 1 by the relative maximum values for which the machine has been furnished.

On the other hand, PlanetLab data traces cover the mean CPU utilization of more than 1 000 VMs, as shown in Table 5. These VMs are sampled over a 5-minute interval on 10 different days from 03/03/2011 to 20/04/2011. During the simulations, each VM is randomly assigned a workload trace from one of the VMs from the corresponding time. The workload makes initiating VMs with a real configuration possible, while CPU utilization of initiated VMs according to the PlanetLab dataset changes over time, similar to real VMs [34].

5.2 Experimental Setup

For proper validation and experimentation, it is always recommended to use real workflow traces. Google offers a Cloud Platform (GCP) suite of Cloud Computing services deployed on the same infrastructure used by Google Search, Gmail and YouTube. Besides the management tools, GCP offers integrated and innovative services such as computing, data storage and analytics and machine learning. In this research work, four heterogeneous VMs are created for parallel execution and validating the performance of ETSA-LP in a cloud environment, as shown in Table 6. All four VMs have diverse configurations to work in a distributed manner

Attribute Name	Datatype
start time	integer
end time	integer
job ID	integer
task index	integer
machine ID	integer
CPU rate	float
canonical memory usage	float
assigned memory usage	float
unmapped page cache	float
total page cache	float

Attribute Name	Datatype
max memory usage	float
disk IO time	float
local diskspace usage	float
maximum CPU rate	float
maximum disk IO time	float
cycles per instruction	float
memory access per instruction	float
sample portion	float
aggregation type	boolean
sampled CPU usage	float

Table 4. Google Cluster: Workload information (CPU and memory traces)

for executing applications. As observed practically and depicted in Figure 3, the average load changes at different time intervals based on the accomplished process.

The experimental evaluation of the proposed ETSA-LP prediction approach is highlighted in this section. As we discussed earlier, thorough testing has been done

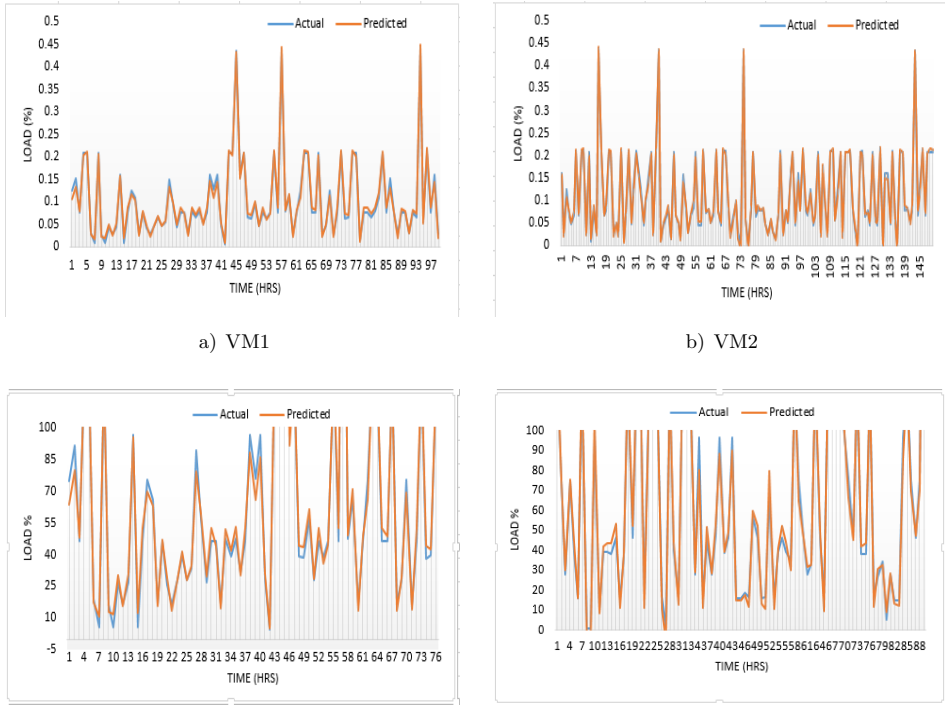
Date	No. of VMs	Mean [%]	SD [%]
03/03/2011	1 052	12.31	17.09
06/03/2011	898	11.44	12.83
09/03/2011	1 061	10.70	15.57
22/03/2011	1 516	9.26	12.78
25/03/2011	1 078	10.56	14.14
03/04/2011	1 463	12.39	16.55
09/04/2011	1 358	11.12	15.09
11/04/2011	1 233	11.56	15.07
12/04/2011	1 054	11.54	15.15
20/04/2011	1 033	10.43	15.21

Table 5. PlanetLab: Workload information

to assess the efficiency of the proposed approach on Google cluster and PlanetLab traces.

VMs	VM Name	Load Type	Avg Load [%]	SD [%]
VM1	Gcpvm1	Low	38.31	3.32
VM2	Gcpvm2	Low	35.45	12.78
VM3	Gcpvm3	High	78.71	6.90
VM4	Gcpvm4	High	72.56	14.55

Table 6. Selected VMs and their average load



a) VM1 b) VM2

c) VM3 d) VM4

Figure 3. Actual and predicted load of different VMs

5.3 Evaluation of Base Models and ETSA-LP

Firstly, the performance of five machine learning base models, statistical ARIMA, ANN, SVM, LSTM and ES is dignified and then with their resulting best predictions, the ensemble model is evaluated. Figure 4 depicts the comparative results

of the five base models tested on the Google cluster for CPU utilization. Similarly, Figure 5 represents the results of base models and the proposed model tested on the Google cluster for memory utilization. On the other hand, the experimentation results conducted using PlanetLab traces for CPU utilization are highlighted in Figure 6. Likewise, memory utilization metrics tested on PlanetLab traces are shown in Figure 7.

As per the experiments, it has been observed that an individual model offers a different performance concerning any evaluation metrics due to its dependency on the tested dataset type. If there is a change in the dataset, the values of the performance metrics also vary. For instance, when applied to the Google traces dataset, SVM beats the other four time-series-based models in accuracy (91.34 %). But, it performs less with accuracy (78.70 %) for the PlanetLab dataset. Therefore, these base time-series models are combined based on the proposed ensemble algorithm discussed in Section 4.1 to boost the performance. The proposed ensemble model ETSA-LP gives the best accuracy (97.45 %) among all other base models with a total execution time 0.39 ms for the Google traces dataset. Similarly, in the PlanetLab dataset, ETSA-LP gives superior accuracy (95.78 %) among the existing models with a total execution time 0.40 ms. The overall accuracy is improved by approximately 3 % and execution time is reduced by 12.2 %. Also, the RMSE value for ES is maximum (2.82 %) whereas ETSA-LP has a minimum error value (0.39 %). Hence, it is substantiated that the proposed ETSA-LP ensemble approach performs better than the existing individual time-series-based models.

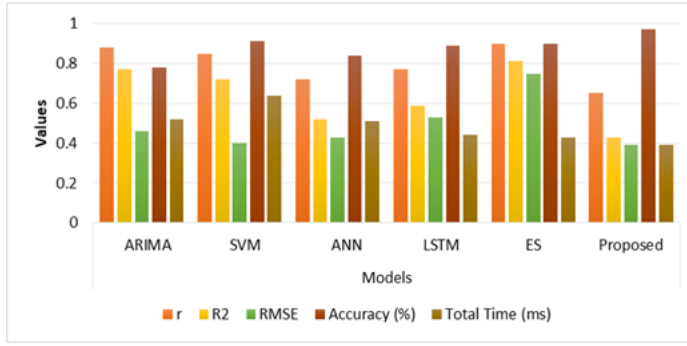


Figure 4. Performance metrics of different models on Google traces (CPU)

5.4 Average CPU and Memory Utilization

Figure 8 a) illustrates a VM's predicted CPU and memory usage during training models on the Google traces and Figure 8 b) corresponds to the utilization deployed on the PlanetLab dataset. In this proposed work, an effort has been made to assess all possible effects of load prediction with low or high CPU and memory usage

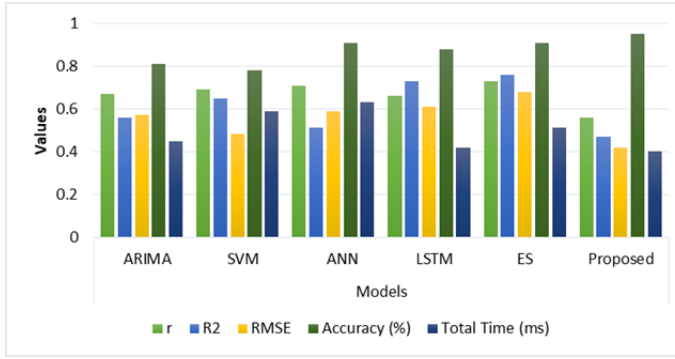


Figure 5. Performance metrics of different models on Google traces (Memory)

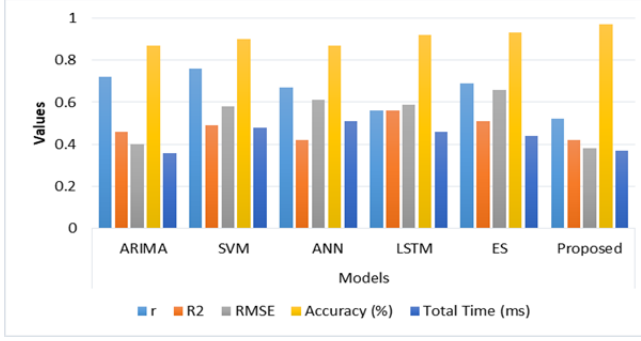


Figure 6. Performance metrics of different models on PlanetLab (CPU)

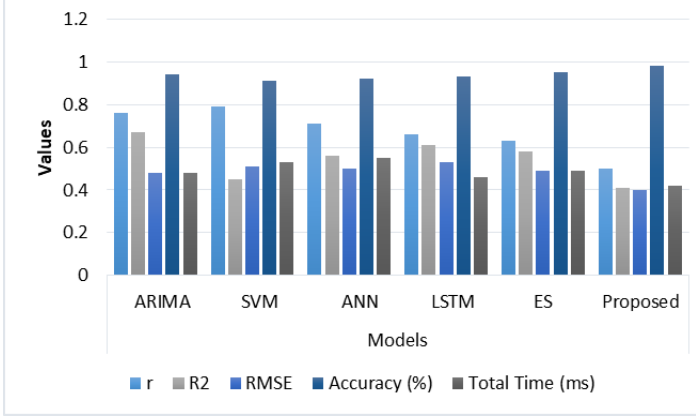
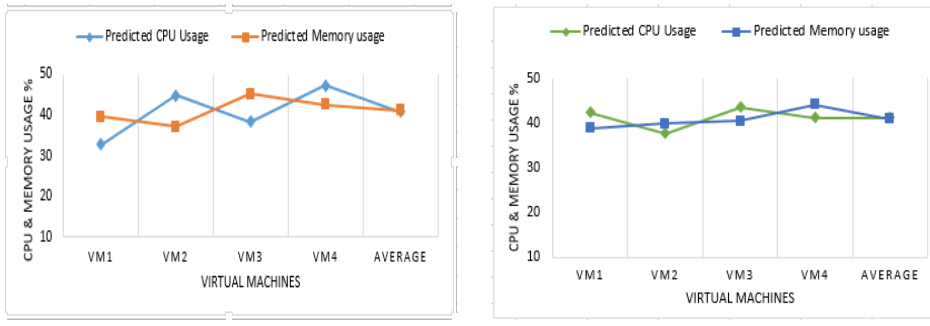


Figure 7. Performance metrics of different models on PlanetLab (Memory)

utilization factors that can directly or indirectly affect the performance of the recommended ensemble approach For instance, in the case of Google traces, the predicted CPU usage of VM1 (31 %) is lower than its memory usage (39 %). Similarly, the predicted CPU usage of VM1 (42 %) is higher than its memory usage (37 %) in the PlanetLab dataset. The results also vary as the time intervals change, as shown in Figure 9. With the difference of five minutes interval, the CPU and memory usage goes high or low according to the number of user requests. The results suggest that with prior knowledge of resources to be utilized for an application, the issue of an under-provisioned host/VM can be resolved and resource-task mapping can be done efficiently.



a) Google traces b) PlanetLab

Figure 8. Predicted CPU and Memory usage

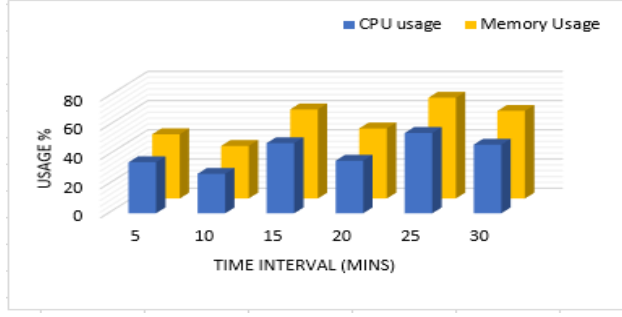
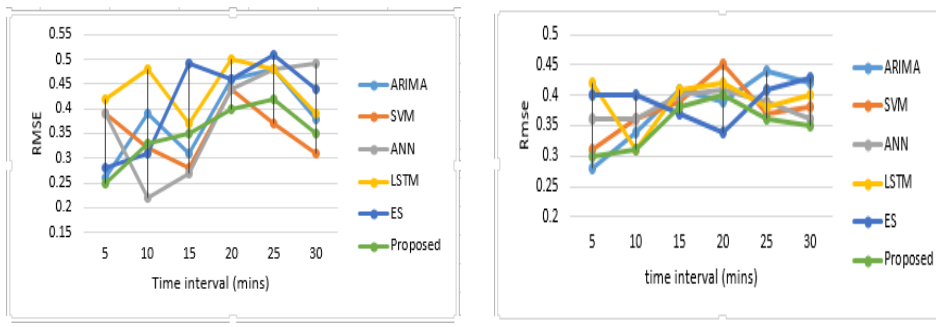


Figure 9. Average CPU and Memory utilization of VMs at different time intervals

5.5 RMSE and Accuracy

Figure 10 compares the mean error RMSE of different VMs at different time intervals. Similarly, the accuracy comparison of other models with the proposed is

demonstrated in Figure 11. The proposed ensemble model ETSA-LP gives the best and higher accuracy among all other base models for all four VMs. The overall accuracy is improved and execution time is also reduced. Also, the RMSE value for ETSA-LP has a minimum error value in comparison with other base models for all VMs. The results recommend that if the load can be effectively expected with former awareness of resource utilization, the over-utilization or under-utilization of resources can be either ceased or handled commendably.



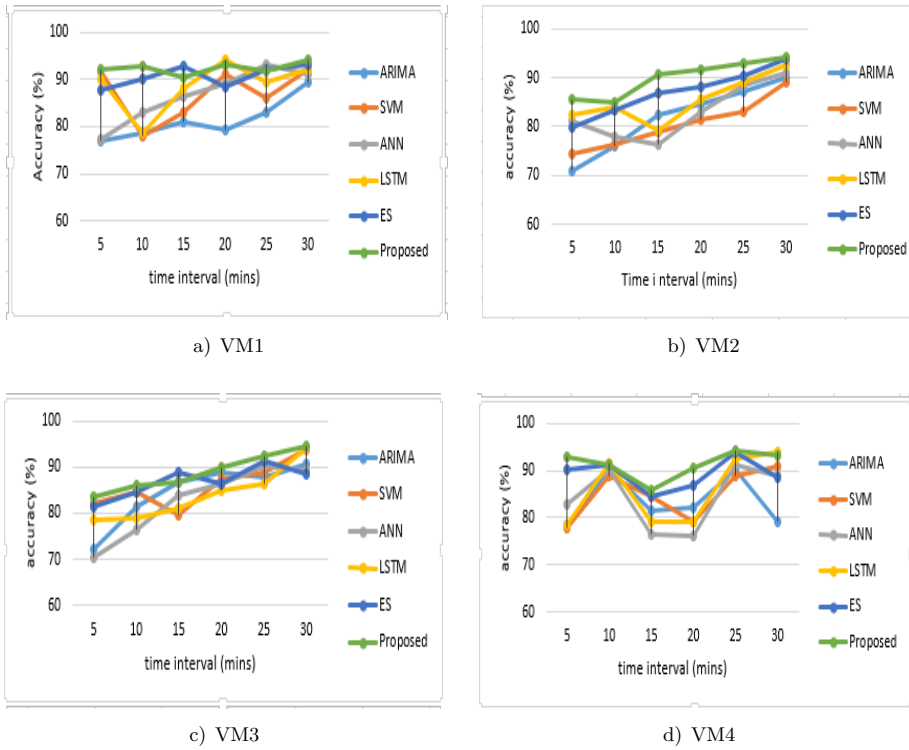
a) VM1 b) VM2

c) VM3 d) VM4

Figure 10. RMSE comparison of different VMs at different time intervals

5.6 Comparison with Existing Approaches

The comparison of the proposed approach ETSA-LP with the existing approaches, namely, LA (Learning Automata), ELM (Extreme Learning Machine) and SVR (Support Vector Regression) is shown in Figure 12, in terms of RMSE and accuracy. For instance, in Figure 12 a), ELM has the highest error rate, whereas the proposed ETSA-LP gives a minimum error value compared to existing approaches. Moreover, ETSA-LP outperforms existing models with the highest accuracy for all the VMs,



a) VM1 b) VM2

c) VM3 d) VM4

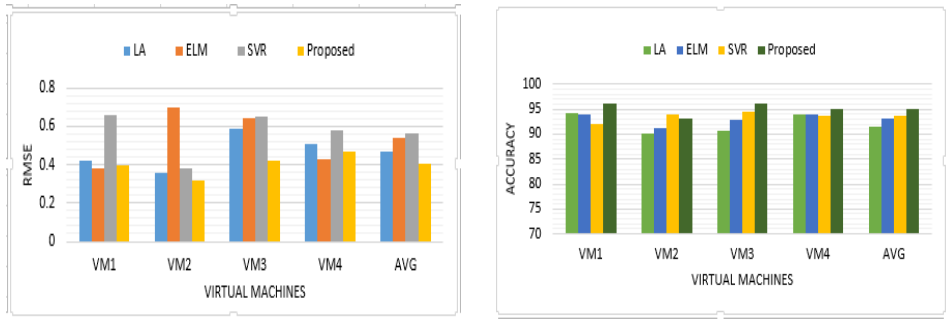
Figure 11. Accuracy comparison of different VMs at different time intervals

as depicted in Figure 12 b). As can be seen, the proposed ensemble approach has less error rate and the best accuracy on all four VMs as compared to the existing ensemble approaches.

6 CONCLUSION AND FUTURE WORK

The primary objective of this research was to compare the performance of existing statistical models with a novel ensemble model for forecasting load using time-series analysis techniques. Another objective of this paper was to systematically evaluate the CPU and memory utilization with dynamic load prediction as a part of the ensemble model. Overall, the ensemble model ETSA-LP is expected to perform better compared to the existing models for the following reasons:

- First, the best performance in terms of error was found from the ETSA-LP due to the reason that the ARIMA and other models are perhaps not able to capture the non-linearities present in the time-series data. However, another reason could be that the ANN possesses several weights and parameters, whereas the



a) RMSE b) Accuracy

Figure 12. Comparison of existing and proposed approach

ARIMA model includes only three parameters. Therefore, the novel ensemble model ETSA-LP performed better as it gave more weight to the accurate model predictions than the less accurate ones and dynamically adjusted its weights.

- Another likely reason behind achieving a substantial boost in performance through proposed ETSA-LP model is the design of its training procedures. After getting the best model configurations from individual models, we trained all unique models several times to obtain the best value of their objective function. The ensemble model dynamically adjusted its weights for calibrating the models using a proposed exponential-weighted algorithm.

Although the proposed ensemble approach performs relatively better than other statistical and neural approaches and is reproducible, this research work has a few implications for future research and data analytics which are as follows:

- It may be expensive to train neural networks and ensemble models. Therefore, it may require model retraining at regular intervals, primarily when more recent data are generated over time.
- Most of the existing load prediction schemes ignore network resources and bandwidth, leaving an opportunity to improve the adequate bandwidth of networks on clusters running parallel applications [35]. For this reason, the proposed load prediction technique could also be deployed to enhance performance and utilization in cluster environments regarding network, CPU, disk and memory I/O resources.
- In addition to weighted ensembles, bagging or bootstrap aggregating also exists, which makes decisions based on the aggregated results of the sampled decision trees [5]. Still, another possibility for future research is to extend univariate time-series forecasting to multi-variate time-series forecasting.
- Recently, a new data-driven paradigm, and termed digital twin (DT) has emerged receiving increasing attention in Cloud. The DT aims to perform continuous

monitoring and proactive maintenance through continuous data from physical to virtual entities [36]. Therefore, the proposed approach can be simulated using the real-time DT, increasing reliability, and maintenance and extending its service life.

- At last, but not least, this effective load prediction strategy can be deployed in autonomic scaling. Auto-Scaling is a technique that automatically scales up and down the resources according to the predicted load [7]. Also, it will ensure lower operational costs in case of resource management resolutions in Cloud Computing environments.

7 STATEMENTS AND DECLARATIONS

I agree with the following statements and declare that this submission follows the policies as outlined in the Guide for Authors and in the Ethical Statement:

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests: The authors have no relevant financial or non-financial interests to disclose.

Ethics approval: This material is the authors' own original work, which has not been previously published elsewhere. The paper is not currently being considered for publication elsewhere. The paper reflects the author's own research and analysis in a truthful and complete manner. The paper properly credits the meaningful contributions of co-authors and co-researchers. The results are appropriately placed in the context of prior and existing research. All sources used are properly disclosed and indicated by giving proper references. All authors have been personally and actively involved in substantial work leading to the paper and will take public responsibility for its content.

Data Availability: Data sharing does not apply to this article as no datasets were generated or analyzed during the current study.

Authors' contributions: All authors contributed to the study's conception and design. Material preparation, data collection, and analysis were performed by Shveta Verma and Dr. Anju Bala. The first draft of the manuscript was written by Shveta Verma and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

REFERENCES

[1] Verma, S.—Bala, A.: A Review: Intelligent Load Prediction Techniques for CloudIoT. Proceedings of the Third International Conference on Advanced Informatics for Computing Research (ICAICR '19), 2019, doi: 10.1145/3339311.3339342.

[2] Cao, J.—Fu, J.—Li, M.—Chen, J.: CPU Load Prediction for Cloud Environment Based on a Dynamic Ensemble Model. Software: Practice and Experience, Vol. 44, 2014, No. 7, pp. 793–804, doi: 10.1002/spe.2231.

[3] Ajila, S. A.—Bankole, A. A.: Using Machine Learning Algorithms for Cloud Client Prediction Models in a Web VM Resource Provisioning Environment. Transactions on Machine Learning and Artificial Intelligence, Vol. 4, 2016, No. 1, pp. 28–51, doi: 10.14738/tmlai.41.1690.

[4] Kaur, G.—Bala, A.—Chana, I.: An Intelligent Regressive Ensemble Approach for Predicting Resource Usage in Cloud Computing. Journal of Parallel and Distributed Computing, Vol. 123, 2019, pp. 1–12, doi: 10.1016/j.jpdc.2018.08.008.

[5] Kaushik, S.—Choudhury, A.—Sheron, P. K.—Dasgupta, N.—Natarajan, S.—Pickett, L. A.—Dutt, V.: AI in Healthcare: Time-Series Forecasting Using Statistical, Neural, and Ensemble Architectures. Frontiers in Big Data, Vol. 3, 2020, Art. No. 4, doi: 10.3389/fdata.2020.00004.

[6] Kumar, J.—Singh, A. K.—Buyya, R.: Ensemble Learning Based Predictive Framework for Virtual Machine Resource Request Prediction. Neurocomputing, Vol. 397, 2020, pp. 20–30, doi: 10.1016/j.neucom.2020.02.014.

[7] Verma, S.—Bala, A.: Auto-Scaling Techniques for IoT-Based Cloud Applications: A Review. Cluster Computing, Vol. 24, 2021, No. 3, pp. 2425–2459, doi: 10.1007/s10586-021-03265-9.

[8] Jeddi, S.—Sharifian, S.: A Hybrid Wavelet Decomposer and GMDH-ELM Ensemble Model for Network Function Virtualization Workload Forecasting in Cloud Computing. Applied Soft Computing, Vol. 88, 2020, Art. No. 105940, doi: 10.1016/j.asoc.2019.105940.

[9] Chen, J.—Wang, Y.: A Resource Demand Prediction Method Based on EEMD in Cloud Computing. Procedia Computer Science, Vol. 131, 2018, pp. 116–123, doi: 10.1016/j.procs.2018.04.193.

[10] Nguyen, H. M.—Kalra, G.—Jun, T. J.—Woo, S.—Kim, D.: ESNemble: An Echo State Network-Based Ensemble for Workload Prediction and Resource Allocation of Web Applications in the Cloud. The Journal of Supercomputing, Vol. 75, 2019, No. 10, pp. 6303–6323, doi: 10.1007/s11227-019-02851-4.

[11] Tofighy, S.—Rahmanian, A. A.—Ghobaei-Arani, M.: An Ensemble CPU Load Prediction Algorithm Using a Bayesian Information Criterion and Smooth Filters in a Cloud Computing Environment. Software: Practice and Experience, Vol. 48, 2018, No. 12, pp. 2257–2277, doi: 10.1002/spe.2641.

[12] Shaw, R.—Howley, E.—Barrett, E.: An Intelligent Ensemble Learning Approach for Energy Efficient and Interference Aware Dynamic Virtual Machine Consolidation. Simulation Modelling Practice and Theory, Vol. 102, 2020, Art. No. 101992, doi: 10.1016/j.simpat.2019.101992.

[13] Xiong, C.—Guan, Y.: A Cloud Computing Load Prediction Hybrid Model with Adaptive Weight Strategy. Signal, Image and Video Processing, Vol. 17, 2023, No. 12, pp. 2101–2109, doi: 10.1007/s11760-022-02424-8.

[14] Belouch, M.—Hadaj, S. E.: Comparison of Ensemble Learning Methods Applied to Network Intrusion Detection. Proceedings of the Second International Con-

ference on Internet of Things, Data and Cloud Computing (ICC '17), 2017, doi: 10.1145/3018896.3065830.

[15] Rahmanian, A. A.—Ghobaei-Arani, M.—Tofighy, S.: A Learning Automata-Based Ensemble Resource Usage Prediction Algorithm for Cloud Computing Environment. Future Generation Computer Systems, Vol. 79, 2018, pp. 54–71, doi: 10.1016/j.future.2017.09.049.

[16] Ghobaei-Arani, M.—Rahmanian, A. A.—Shamsi, M.—Rasouli-Kenari, A.: A Learning-Based Approach for Virtual Machine Placement in Cloud Data Centers. International Journal of Communication Systems, Vol. 31, 2018, No. 8, Art. No. e3537, doi: 10.1002/dac.3537.

[17] Mauldin, T.—Ngu, A. H.—Metsis, V.—Canby, M. E.—Tesic, J.: Experimentation and Analysis of Ensemble Deep Learning in IoT Applications. Open Journal of Internet of Things (OJIOT), Vol. 5, 2019, No. 1, pp. 133–149.

[18] Toumi, H.—Brahmi, Z.—Gammoudi, M. M.: RTSLPS: Real Time Server Load Prediction System for the Ever-Changing Cloud Computing Environment. Journal of King Saud University – Computer and Information Sciences, Vol. 34, 2022, No. 2, pp. 342–353, doi: 10.1016/j.jksuci.2019.12.004.

[19] Tuli, S.—Basumatary, N.—Gill, S. S.—Kahani, M.—Arya, R. C.—Wander, G. S.—Buyya, R.: HealthFog: An Ensemble Deep Learning Based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in Integrated IoT and Fog Computing Environments. Future Generation Computer Systems, Vol. 104, 2020, pp. 187–200, doi: 10.1016/j.future.2019.10.043.

[20] Belgacem, A.—Mahmoudi, S.—Kihl, M.: Intelligent Multi-Agent Reinforcement Learning Model for Resources Allocation in Cloud Computing. Journal of King Saud University – Computer and Information Sciences, Vol. 34, 2022, No. 6, pp. 2391–2404, doi: 10.1016/j.jksuci.2022.03.016.

[21] Rashida, S. Y.—Sabaei, M.—Ebadzadeh, M. M.—Rahmani, A. M.: An Intelligent Approach for Predicting Resource Usage by Combining Decomposition Techniques with NFTS Network. Cluster Computing, Vol. 23, 2020, No. 4, pp. 3435–3460, doi: 10.1007/s10586-020-03099-x.

[22] Bi, J.—Li, S.—Yuan, H.—Zhou, M. C.: Integrated Deep Learning Method for Workload and Resource Prediction in Cloud Systems. Neurocomputing, Vol. 424, 2021, pp. 35–48, doi: 10.1016/j.neucom.2020.11.011.

[23] Ngo, N. T.—Pham, A. D.—Truong, T. T. H.—Truong, N. S.—Huynh, N. T.—Pham, T. M.: An Ensemble Machine Learning Model for Enhancing the Prediction Accuracy of Energy Consumption in Buildings. Arabian Journal for Science and Engineering, Vol. 47, 2022, No. 4, pp. 4105–4117, doi: 10.1007/s13369-021-05927-7.

[24] Wang, S.—Zhu, F.—Yao, Y.—Tang, W.—Xiao, Y.—Xiong, S.: A Computing Resources Prediction Approach Based on Ensemble Learning for Complex System Simulation in Cloud Environment. Simulation Modelling Practice and Theory, Vol. 107, 2021, Art. No. 102202, doi: 10.1016/j.simpat.2020.102202.

[25] Gunasekaran, J. R.—Mishra, C. S.—Thinakaran, P.—Kandemir, M. T.—Das, C. R.: Cocktail: Leveraging Ensemble Learning for Optimized Model Serving

in Public Cloud. CoRR, 2021, doi: 10.48550/arXiv.2106.05345.

[26] Yazdanian, P.—Sharifian, S.: E2LG: A Multiscale Ensemble of LSTM/GAN Deep Learning Architecture for Multistep-Ahead Cloud Workload Prediction. The Journal of Supercomputing, Vol. 77, 2021, No. 10, pp. 11052–11082, doi: 10.1007/s11227-021-03723-6.

[27] Bao, L.—Yang, J.—Zhang, Z.—Liu, W.—Chen, J.—Wu, C.: On Accurate Prediction of Cloud Workloads with Adaptive Pattern Mining. The Journal of Supercomputing, Vol. 79, 2023, No. 1, pp. 160–187, doi: 10.1007/s11227-022-04647-5.

[28] Kumar, J.—Singh, A. K.: Workload Prediction in Cloud Using Artificial Neural Network and Adaptive Differential Evolution. Future Generation Computer Systems, Vol. 81, 2018, pp. 41–52, doi: 10.1016/j.future.2017.10.047.

[29] Chandini, M. S.—Pushpalatha, R.—Boraiah, R.: A Brief Study on Prediction of Load in Cloud Environment. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 5, 2016, No. 5, pp. 157–162.

[30] Liu, C.—Liu, C.—Shang, Y.—Chen, S.—Cheng, B.—Chen, J.: An Adaptive Prediction Approach Based on Workload Pattern Discrimination in the Cloud. Journal of Network and Computer Applications, Vol. 80, 2017, pp. 35–44, doi: 10.1016/j.jnca.2016.12.017.

[31] Chen, Z.—Hu, J.—Min, G.—Zomaya, A. Y.—El-Ghazawi, T.: Towards Accurate Prediction for High-Dimensional and Highly-Variable Cloud Workloads with Deep Learning. IEEE Transactions on Parallel and Distributed Systems, Vol. 31, 2020, No. 4, pp. 923–934, doi: 10.1109/TPDS.2019.2953745.

[32] Gupta, N.—Ahuja, N.—Malhotra, S.—Bala, A.—Kaur, G.: Intelligent Heart Disease Prediction in Cloud Environment Through Ensembling. Expert Systems, Vol. 34, 2017, No. 3, Art. No. e12207, doi: 10.1111/exsy.12207.

[33] Wilkes, J.—Reiss, C.: Google Cluster Data. 2011, `https://github.com/google/cluster-data/blob/master/ClusterData2011_2.md` (Accessed 19-Nov-2022).

[34] Beloglazov, A.: PlanetLab Traces. 2011, `https://github.com/beloglazov/planetlab-workload-traces` (Accessed 15-Nov-2022).

[35] Ageed, Z. S.—Mahmood, M. R.—Sadeeq, M. M. A.—Abdulrazzaq, M. B.—Dino, H. I.: Cloud Computing Resources Impacts on Heavy-Load Parallel Processing Approaches. IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 22, 2020, No. 3, pp. 30–41.

[36] Dang, H. V.—Tatipamula, M.—Nguyen, H. X.: Cloud-Based Digital Twinning for Structural Health Monitoring Using Deep Learning. IEEE Transactions on Industrial Informatics, Vol. 18, 2022, No. 6, pp. 3820–3830, doi: 10.1109/TII.2021.3115119.

Shveta Verma is a Research Scholar in the Department of Computer Science and Engineering, Thapar University, Patiala, India. She received her M.Eng. (Master of Engineering) from the Thapar University, Patiala and is pursuing a Ph.D. in the research area of cloud computing from the Thapar University, Patiala.

Anju Bala is working as Associate Professor in the Department of Computer Science and Engineering, Thapar University, Patiala, India. She received her B.Eng. in computer science and engineering and her M.Tech. from the Punjabi University, and her Ph.D. in the research area of cloud computing from the Thapar University, Patiala. She has more than 50 research publications in reputed journals and conferences and guided more than 35 M.Eng. thesis in the same area.