Dynamic Resource Utilization Prediction Model for Cloud Datacenter

Doaa Bliedy, Mohamed H. Khafagy, and Rasha M. Badry

Abstract—**Much research has been done on predicting resource utilization in the cloud to avoid over- and under-provisioning resources. Most existing systems focus on estimating the utilization of one or two resources at most, including memory, CPU, storage, network, or servers dedicated to cloud applications; they do not consider the correlation between resources. A maximum of one or two machine learning algorithms is employed for prediction purposes. Additionally, traditional prediction methods in the cloud provide a one-dimensional output. Most current solutions predict resources, such as memory and CPU utilization, as a single output. Unfortunately, a one-dimensional output in resource provision and usage cannot capture the relationship between the application requirements of different resources, such as CPU, memory, CPU cores, disk, and network; this results in incomplete and inaccurate information and prediction results. Efficient resource allocation and management require predicting several parameters using multivariate state variables. This study presents a multi-resource utilization prediction model that uses several machine learning approaches, such as support vector regression, random forests, MLP regression, neural networks (NN) using Adam and SGD optimizers, and decision tree regression. The prediction model is based on univariate and multivariate time series. Google cluster trace data is used to evaluate the work. Four experiments are executed on the dataset, seeking to predict the resources for different time series interval periods. The various algorithms' mean absolute errors (MAE), root mean square error, mean absolute percentage error and R-squared are compared to determine which technique achieves the lowest error rate. The experimental results have shown that the prediction model yields better accuracy than previous research.**

Index Terms—**cloud computing, resource utilization, prediction, machine learning**

I. Introduction

Effective usage of resources in the cloud enables providers to deliver great performance at a minimal cost. Pay-as-you-go pricing is a common practice among cloud service providers, which allows for cost savings and flexibility for cloud users. The wide range of advancements in cloud computing technologies has led to a considerable rise in cloud users and a growing number of applications that can be used to access various cloud computing services [1].

Many scientific applications use cloud computing services, leading to varying levels of resource utilization [2, 3]. As a result, effective resource management is required to address the changing demands of users. Effective resource management can help with cost reduction, performance enhancement, and resource utilization optimization in a cloud computing context.

Prediction of resource utilization has been thoroughly explored, with a wealth of literature accessible at [3, 4, 5–7]. The accuracy of the resource utilization prediction model, its memory and time requirements, and its ability to handle various resources are all essential factors to consider. Making an accurate resource utilization prediction model is challenging because of the multiple parameters, such as CPU, memory consumption, disk I/O, and network throughput. There may be implicit correlations between memory utilization and CPU consumption, as well as between memory and disk I/O. Finding and predicting the relationship between each resource type is difficult. The predictions' outcomes will not be suitable for practical use in this way. To address this issue, the resource prediction's auto-scaler must manage the multiple indications simultaneously to make accurate scaling decisions.

The article's research contribution can be summed up as follows: A state-of-the-art literature review of the most recent research is provided, covering the analysis of several techniques for predicting resource utilization in cloud computing. A multi-resource utilization prediction model is presented using various machine learning techniques, including neural networks (NN) with Adam and SGD optimizers, MLP regression, random forest, decision tree regression, and support vector regression. The objective was to predict CPU, memory, disk utilization, and disk I/O time more accurately compared to previous traditional methods. The prediction model works for univariate and multivariate time-series input, allowing for the simultaneous prediction of multiple resources. Extensive experiments were conducted on real Google cluster trace data to evaluate and compare the various machine-learning techniques, demonstrating the proposed approach's effectiveness. The comparison of the mean absolute errors achieved by the different methods at different time intervals provided useful insights into selecting the best models.

The following sections comprise the article: Section II briefly summarizes the recent literature pertinent to the proposed work and addresses related work. The suggested model is presented in Section III, along with relevant

Manuscript received December 14, 2023; revised October 2, 2024.

Doaa Bliedy is a researcher at the Department of Information Systems, Faculty of Computers and Artificial Intelligence, Fayoum University, Egypt. (e-mail: db1112@fayoum.edu.eg).

Mohamed H. Khafagy is a professor at the Department of Computer Science, Faculty of Computers and Artificial Intelligence, Fayoum University, Egypt. (e-mail: mhk00@fayoum.edu.eg).

Rasha Badry is an assistant professor at the Department of Information Systems, Faculty of Computers and Artificial Intelligence, Fayoum University, Egypt. (e-mail: rmb01@fayoum.edu.eg).

discussions, figures, and algorithms. The debate and results are provided in Section IV. The article is concluded in Section V.

II. LITERATURE REVIEW

Techniques for predicting cloud resource utilization are well-documented. The associated techniques are described in detail in this section. In [4], the authors have presented a machine learning regression integration method for predicting CPU resource utilization in a scientific application. The suggested method integrates resource utilization with function selection to enhance prediction performance. The model is evaluated on the Cybershake dataset, generated by simulating the application on a workflow management system. The outcomes demonstrate that the suggested model performs more accurately and quickly than existing machine learning regression models because it enhances accuracy by 2% and reduces execution time by 16.2%. The method improves prediction, decreases failures, and enables fault-tolerant scheduling. For cloud users, the scalability of virtualization technology means either excessive demand or insufficient resources over time [3]. The effective use of cloud services becomes much more challenging as a result. The resource utilization model depends on time and is impacted by cloud resource usage trends. In [3], the researchers estimate the CPU load of numerous virtual machines (VMs) obtained from the CoMon project dataset by combining a genetic algorithm with a learning automata (LA) theory-based cloud resource utilization prediction algorithm. The CoMon project [8] includes CPU utilization of more than a thousand VMs in intervals of 5 minutes for 24 hours. The algorithm uses prediction models to determine the weights for individual models. The suggested algorithm is tested by predicting the load on several virtual machines. The outcomes demonstrate that the proposed algorithm outperforms other prediction systems because it achieves the lowest RMSD error, which accounts for 8.77169.

In [6], an adaptive method for workload prediction using SVM and LR is presented. This method first divides workloads into numerous categories. Additionally, it automatically assigns various prediction models based on the priority of the jobs, the speed of workload change, and the characteristics of the workload. The Google Cluster trace is used to evaluate the proposed method. The experimental results have shown that when compared to the time-series prediction methods (Autoregressive Integrated Moving Average (ARIMA), Support Vector Machines (SVM), and Linear Regression (LR), the proposed method reduces the platform's cumulative relative prediction errors by 29.06%, 8.42%, and 40.86%, respectively. Automated resource provisioning adjusts available resources to match service needs. The more accurate the prediction model is, the greater the reduction in power consumption and the higher the assurance in SLA and QoS, especially for services with strict QoS requirements regarding latency or response time.

The authors of [7] proposed an approach that employs an SVM regression model to forecast the average hourly load of a distributed server during a 24-hour test interval based on historical data and estimate the necessary number of resources. The SVM-based forecasting techniques are compared with other forecasting methods, namely those based on last-value, moving average, and linear regression. In addition to these basic methods, the SVM-based forecasting models are compared and evaluated using three different kernel functions (SVM with a polynomial kernel, SVM with an RBF kernel, and SVM with a normalized polynomial kernel). The suggested forecasting model evaluation uses real online service logs from Complutense University of Madrid. According to experimental results, prediction errors (MAE, MSE, and RMSE) are fewer in all cases when utilizing kernel functions in SVM-based forecasting models than when using the three fundamental methodologies (SVM based on last-value, moving average, and linear regression). The SVM algorithm and other machine learning techniques, such as Naive Bayes (NB), k-nearest neighbor (K-NN), decision trees (DT), logistic regression, and random forest (RF) are proven for their efficiency and accuracy in other fields, such as detecting Deep fake videos and tracing the origin of metastatic lung cancer tissues [20, 25].

In [9], an adaptive model called Long Short Term Memory (LSTM) is presented to predict CPU load. It predicts the average load in advance at consecutive future intervals. The model's performance is evaluated on two real workloads: a workload trace on UNIX systems collected by Dinda and a workload trace from the Google data center. The artificial neural networks (ANN), the Bayes model, the autoregressive (AR) model, the PSR+EA-GMDH method, and the echo state networks (ESN) are compared with the proposed model. The results have demonstrated that, in comparison to other models, the suggested approach is more accurate on both datasets. The presented model and the ESN model both vastly outperform the PSR+EAGMDH and the Bayes methods, and using the LSTM model is slightly better than the ESN method. The ESN and LSTM models can use historical data to determine long-term dependencies.

The authors of [10] presented a model for predicting workloads using neural networks and an adaptive differential evolution algorithm. This model enables administrators to discover the potential issues with the resource reservation plan and alter it as needed. The algorithm then decides if resources are over or under-provisioned. The knowledge extracted during this process is then utilized to examine characteristics of resource utilization. The suggested solution is examined on OpenStack using Wikipedia server traffic data. The neural network and other machine learning algorithms, including linear regression and RepTree, are compared. The RepTree method outperformed the neural network by 7% and learns, but the neural network model performs better in the long term. Scalability, a crucial cloud computing element, is achieved through efficient resource scheduling. Determining whether a resource reservation strategy can be developed and implemented for optimal resource scheduling is critical. Such a plan can distribute additional resources while keeping enough available. A neural network technique is also used in this study [22–24] to predict short-term power loads, wind speed, and stock market prices. The fault tolerance and robustness of neural networks allow them to predict many complex nonlinear time series systems accurately. Using neural networks can help handle noise data and disturbances in power systems, leading to improved prediction stability.

In [11], neural networks and a self-adaptive differential evolution technique are used to predict resource utilization. The technique selects the most suitable crossovers and mutations. HTTP traces from NASA and Saskatchewan servers are used to evaluate the method. Considerable improvements are found when comparing the model with other prediction models built based on the well-known backpropagation learning algorithm. The model achieved a reduction in error of up to 168 times.

In [12], researchers introduced an LSTM network algorithm to predict workload in cloud datacenters. The experiments were performed on three datasets: HTTP traces of the NASA server, the Calgary server, and the Saskatchewan server, which were gathered and examined by [13]. For HTTP traces from the NASA server, Calgary server, and Saskatchewan server, the minimum mean squared errors were 4.79×10^{-3}, 3.42×10^{-3}, and 3.17×10^{-3}, respectively. The results have demonstrated that the model achieved remarkable prediction accuracy by reducing the mean squared error to 3.17×10^{-3}. To construct models based on various workload attributes, most clouds don't employ more than user-specified resource use thresholds to offer automatic scaling. Fewer jobs can estimate resource needs by analyzing multiple indicators at once.

In [14], the authors utilized a Long Short-Term Memory (LSTM) neural network to predict resource utilization with multivariate time series data. The model employed is the Multivariate Fuzzy LSTM (MF-LSTM), a novel cloud-active automated scaling system that integrates several mechanisms. To select appropriate inputs, the correlation between various metrics is assessed. It is recommended to use a fuzzification technique to reduce the fluctuation of monitoring data. The authors evaluated their model using Google Trace data. The results show that the CPU and memory predictions using MF-LSTM have Mean Absolute Error (MAE) values of 0.3221 and 0.0303, respectively. Time series forecasting (TSF) is a common research task in various domains, such as medical, transportation, environment, network detection, and finance [27].

In [15], the authors developed a scheduling algorithm based on a prediction model to address the issue of peak loads, which can lead to scheduling errors and reduce the energy efficiency of the algorithm. It is challenging for any predictive model to accurately predict the future resource usage of a data center based solely on initial data. Therefore, the best scheduling algorithm is one that can accurately predict data to handle complex scheduling scenarios while ensuring Quality of Service (QoS) and avoiding Service Level Agreement (SLA) violations. The authors compared the accuracy of their scheduling algorithm with the round-robin (RR) scheduling algorithm, the Minimum Migration Time (MMT) scheduling algorithm, and the First-Fit (FF) scheduling algorithm using the Google trace dataset. The results of the proposed approach showed that the proposed algorithm utilized more CPU and memory compared to the other three algorithms.

In [16], a model is presented to predict the execution time of Hadoop Map-Reduce applications in a private cloud environment using a regression-based performance model. Cloud computing Map-Reduce packages can optimize the allocation of resources and complete Map Reduce jobs within a specified timeframe. Users of cloud services need to estimate the resources required to complete tasks in modern systems. The proposed framework predicts job completion times using a scale-out strategy. The datasets are randomly generated using tools and programs such as the Random Writer Tool and the TeraGen program. The results demonstrated that the model achieved a high accuracy rate of 99%.

In [17], an online learning approach for multivariate resource usage prediction models is proposed using the Levenberg-Marquardt and gradient descent methods. The predicted resources are CPU usage for seven and twenty days. The framework is evaluated using the PlanetLab workload trace and the Google cluster trace. A comparison between the learning abilities of the ARIMA and BLSTM models demonstrates that the BLSTM model performs significantly better. Sparse BLSTM is presented to address the challenge of adapting many parameters in BLSTM. A concept tree is created to help identify the parameters needing removal. Predictions of adapted sparse models are comparable to those of adapted dense models. When comparing the adaptation times for dense and sparse models, it can be seen that sparse real-time adaptations are faster by 50–60% in the pruned model.

In [18], the authors suggested a multi-objective load-balancing approach integrated with a prediction model called the OP-MLB framework for elastic resource management at a cloud data centre. They used neural networks customized with an adaptive evolutionary algorithm to predict cloud resources. Multi-objective load balancing is achieved through proactive VM placement and migration, where VMs are allocated based on maximum resource utilization and minimal power and communication costs. The presented framework is evaluated on three real benchmark datasets: Google Cluster Data (GCD), PlanetLab VMs (PL), and the Bitsbrain (BB) dataset. The lowest RMSE error score of the proposed prediction approach for a prediction interval of 5 minutes on the three workloads is 0.0005 for CPU resources.

In [19], a hybrid LSTM (Convolutional Neural Network and Long Short-Term Memory) model for analyzing multivariate workloads is presented. The main goal of this model is to extract the complex features of the VM usage components and model temporal information about the irregular trends in the time series components. Bitbrains data is used to evaluate the presented model. The suggested and alternative prediction models, including ARIMA-LSTM, VAR-GRU, and VAR-MLP, are compared. The proposed model's accuracy rate (which was improved from 3.8% to 10.9%) and the error rate (which decreased to 7% from 8.5%) are better than other models, according to the results.

The authors of [22] suggested a short-term load prediction approach using a Group Method of Data Handling (GMDH)-type neural network. Combining the GMDH-type neural network could lead to accurate short-term load prediction in power systems. The neural network with its excellent fault tolerance and robustness analyzes nonlinear historical load data and other pertinent factors to learn patterns and forecast the future. Furthermore, simulation experiments were conducted to validate the effectiveness of the presented method, which fully demonstrated its ability to accurately predict short-term load in power systems, resulting in better operational planning and decision-making, as well as improved power system technical and economic performance. Overall, using GMDH-type neural networks for short-term load forecasting proved to be effective in improving the operational efficiency and reliability of modern power systems.

In [23], the authors presented an accurate, stable, and efficient wind speed prediction method integrated with an error correction mechanism to address the strong randomness and volatility of wind speed data. They utilized a convolutional neural network (CNN) for prediction and optimized the data weights using the PSO algorithm. Singular spectrum and wavelet analysis were employed to denoise and process the wind speed data. This study evaluated the effectiveness of the prediction method using two wind speed datasets from a wind farm in Jiangsu Province, China. The wind speed prediction model proposed in this research outperforms previous models in short-term prediction, as demonstrated by testing results on the two datasets.

A study by [24] employed artificial neural networks to predict stock prices during the COVID-19 pandemic, specifically focusing on the Indonesia Stock Exchange. They trained the ANN with historical stock data and various market indicators to get more accurate predictive results and capture complex patterns. The outcomes showed that the ANN-based model performed better than conventional statistical techniques, providing more accurate stock movement forecasts up to a maximum of 98.8%. There are recorded MAPE values less than or equal to 10%. This could be helpful for investors navigating erratic market conditions during the epidemic.

The authors of [25] developed a hybrid technique for identifying deep fakes in videos. They used machine learning, deep learning, and YOLO-V3 techniques to detect and extract features from faces in videos. An ensemble of machine learning classifiers is used to detect deep fakes, including support vector machines (SVM), decision trees (DT), k-nearest neighbors (K-NN), and Naïve Bayes (NB). They integrated the Celeb-DF (v2) and Face Forensics++ (FF++) datasets to evaluate the recommended technique. The results show that the recommended method outperforms state-of-the-art techniques with an accuracy of 99.64%. These results imply that the model provides investors navigating the volatile market conditions brought on by the pandemic with useful insights.

In [26], the authors used a Functional Link Neural Network (FLNN) with a hybrid genetic algorithm (GA) and particle swarm optimization (PSO) to develop a multi-resource utilization prediction model. They only utilized one method to forecast CPU and memory resources; their model did not consider disk resource utilization or disk I/O time. They conducted one experiment on the dataset, aiming to predict the resources for a single time series interval period of 5 minutes. The lowest MAE errors achieved in the univariate input case were 0.25 for CPU resources and 0.018 for memory resources. The lowest MAE errors obtained in the multivariate input case were 0.33 for CPU resources and 0.026 for memory resources.

Limitations of Previous Work

- Most research on cloud resource prediction focuses on predicting cloud resources based on univariate input cases where the prediction is based on a single input and single output. There is relatively little work exploring multivariate input cases, where multiple input variables are used simultaneously to enhance prediction accuracy. Addressing this gap could lead to more robust and comprehensive resource prediction models that better reflect the dynamic nature of cloud environments.
- They focused on forecasting CPU and memory resources using just one or two techniques without taking disk utilization and disk I/O time into account. This strategy reduces the efficacy of their models since it ignores important elements that affect system performance as a whole. There is a need for incorporating disk-related metrics with CPU and RAM, employing advanced or hybrid modeling methodologies for a more holistic approach to resource management in cloud environments, in order to build more thorough and accurate resource predictions.
- They executed one or two experiments at most to evaluate their work, seeking to predict the resources for only one or two-time series intervals. This narrow approach restricts the generalizability of their models, as it does not adequately reflect the diverse and dynamic nature of cloud resource demands over different timeframes.
- Only one or two performance metrics are reported in their experiments, which offers an insufficient assessment of the models' efficacy. This constrained evaluation ignores a thorough comprehension of the models' behavior under diverse circumstances, potentially hiding important features like accuracy, scalability, and robustness. Future studies should include a wider range of performance criteria for a more comprehensive assessment that better captures the advantages and disadvantages of the models in various circumstances.

Despite the number of solutions in the literature, there is still a need for advanced methods with higher accuracy and faster execution times to predict resource utilization in both univariate and multivariate input cases.

Table I provides a comprehensive comparison between the proposed model and several relevant prior models, highlighting important elements such as the models' strengths and weakness. It addresses the drawbacks of each technique, such as computational complexity and scalability concerns, while also emphasizing its benefits, such as accuracy and interpretability.

TABLE I
STRENGTHS AND WEAKNESS OF ALGORITHMS

Ref	Algorithm	Strengths	Weaknesses
[6]	SVM LR	• Linear regression is simple to implement, interpretable • SVM is effective in high-dimensional spaces	- linear regression has limited accuracy with non-linear relationships - SVM requires careful tuning, can be slow for large datasets
[9]	LSTM	• Capturing temporal dependencies, good for sequential data	- Requires large amounts of data, complex to train - Limited generalization to diverse workloads - Limited scope, not scalable to multivariate data
[17]	• Gradient descent(GD) • Levenberg-Marquardt(LM)	• Adaptable to new and dynamic data in cloud	- Requires longer processing time for learning from new data
[26]	• FLGAPSONN (Functional link neural network) with genetic algorithm and particle swam optimization	• Good at feature extraction, can handle spatial data • Leveraging strengths of multiple approaches, improved accuracy	- Not ideal for sequential data, requires extensive data
proposed Model	• Neural network with Adam optimizer ((NN(Adam)) • Neural network with SGD optimizer ((NN(SGD)) • Support Vector regression (SVR) • Random Forest (RF) • Multi-layer Perceptron regression (MLP) • Decision Tree Regression (DTR)	• Capture complex patterns (NN, MLP) • Robust to noise (SVR, RF) • Good with high-dimensional data (SVR) • Feature importance (RF) • Easy interpretation (Decision Tree)	- Computationally intensive (NN, SVR, RF) - Prone to overfitting (NN, MLP, Decision Tree) - Sensitive to tuning (SVR, NN) - Lower interpretability (NN, RF).

This paper presents a multi-resource utilization prediction model that uses several machine learning approaches, not only one. The key strengths of the paper are:

- A prediction model that works for both univariate and multivariate time-series input is proposed, allowing the prediction of multiple resources simultaneously. This approach is more practical than single-resource prediction.
- Extensive experiments were conducted on Google cluster trace data to evaluate and compare the various machine-learning techniques. This demonstrated the effectiveness of the proposed approach.
- A wider range of cloud metrics, such as disk resources and disk I/O, have been included in this paper for better generalizability.
- The comparison of mean absolute error, root mean square error, mean absolute percentage error and R-squared achieved by different techniques for different time intervals provided valuable insights into selecting the best models.

III. MODELS AND METHODS

The proposed model aims to predict multi-resource utilization using machine learning techniques, including neural networks (NN) with Adam and SGD optimizers, MLP regression, random forest, decision tree, and support vector regression. Decreasing the number of training layers in the neural network leads to improved solutions, reducing the error produced. The prediction model is built to predict multi-resources for different periods, including CPU, memory, hard disk, and disk I/O time. The mean absolute error, root mean square error, mean absolute percentage error and R-squared of the various machine learning techniques are compared to determine which achieves the lowest error rate for the different resources. Google cluster trace data is used to evaluate the work.

The primary objective is to choose and integrate several prediction methodologies properly to increase the accuracy of the final forecast. Accuracy can be improved by leveraging each predictive model's benefits and reducing its drawbacks. This section provides details on the suggested algorithm's various elements. drawbacks. This section provides details on the suggested algorithm's various elements.

Proposed Algorithm

Figure 1 depicts the four key modules of the cloud resource forecasting system [5]: the manager, preprocessor, trainer, and forecaster. Furthermore, each module communicates with the others to produce accurate and timely forecasting results. timely forecasting results.

1) Manager or collector module

Through the Manager or Collector module, raw resource monitoring data from VMs is gathered and saved in a repository. Numerous monitoring services for public clouds are currently available, including IBM Cloud Monitoring, Rackspace Monitoring, and Cloud Watch from Amazon Web Services. Users can also configure and set up monitoring tools like Nagios, and Prometheus.

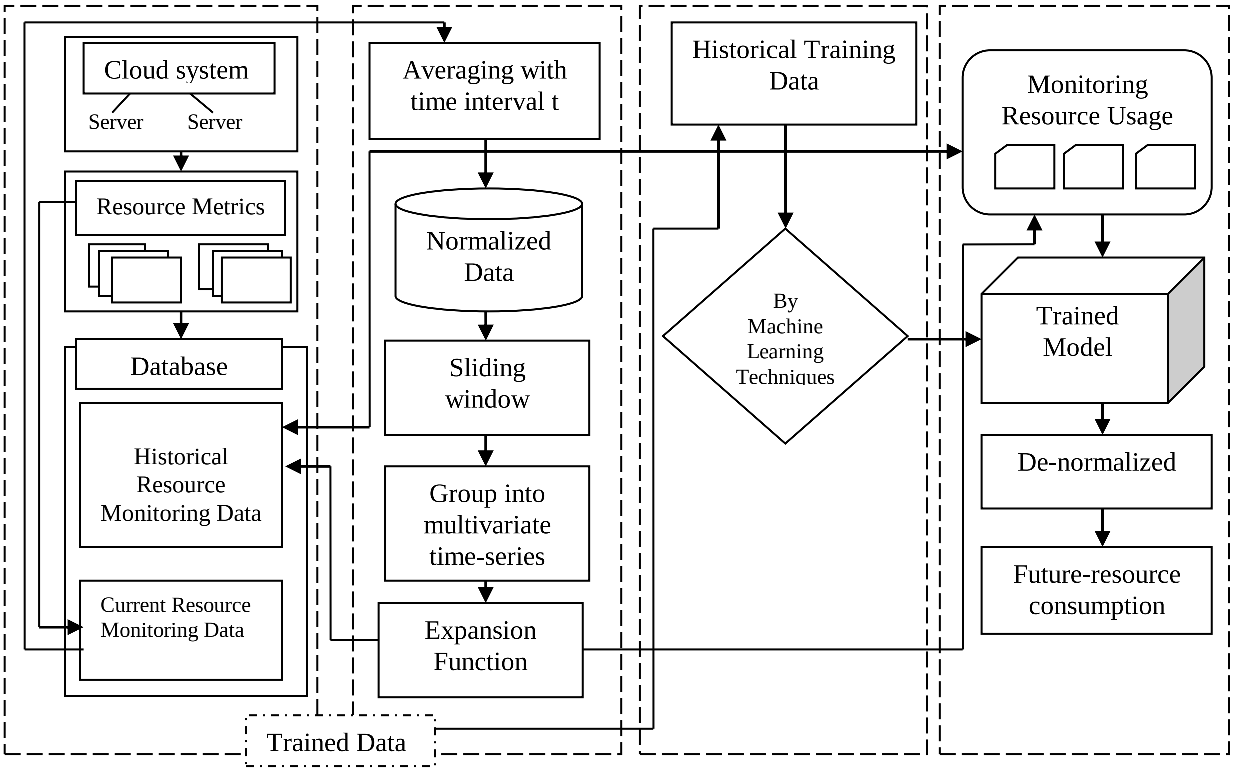
Cloud system
Server
Server
Resource Metrics
Database
Historical Resource Monitoring Data
Current Resource Monitoring Data
Averaging with time interval t
Normalized Data
Sliding window
Group into multivariate time-series
Expansion Function
Trained Data
Historical Training Data
By Machine Learning Techniques
Monitoring Resource Usage
Trained Model
De-normalized
Future-resource consumption

Fig. 1. Architecture of the predictive system

2) Preprocessor module

The function of the preprocessor module is to convert the collected time-series data into supervised data that can be used as input by neural networks and other machine learning techniques. Several mechanisms are implemented to process cloud data, including data normalization, expansion functions, sliding windows, averaging data over a long time, and grouping it into multivariate time series. The output data from this module's preprocessing is entered into the Collector module database as historical resource data, which is utilized to build the prediction model in the Trainer module. The data is also made available to the forecaster to predict resource consumption. As previously mentioned, the goal of the preprocessor is to prepare data for the trainer and forecaster modules. Five mechanisms are deployed in this component. Firstly, the current raw data gathered over a long period is transformed into the corresponding time series with a time interval. The next phase is normalization, which scales a time series in the range of [0, 1]. Then, time series data is transformed into supervised data using the sliding method with a window width of k, which represents the number of values before time t to predict the value at time t. Finally, all resource metric types are grouped into a single multivariate dataset.

3) Trainer module

A learning method is proposed in the trainer module using neural networks and various machine learning techniques. Adam and SGD optimizers are also used to train the network to improve forecast accuracy further. After the training process is complete, the Forecaster module employs the trained model to forecast future resource use.

4) Forecaster module

Values from real-time monitoring data (after the pre-processing process) are used in the Forecaster module as inputs for the trained model to predict new values (i.e., resource consumption) in advance. The resulting outputs are unnormalized into real numbers before being put to use. Initially, raw resource monitoring data from VMs is gathered, and the resource data is collected from Google cluster trace data. After collecting the data, the pre-processing phase begins. The datasets are converted and formatted appropriately at this phase. All null values are eliminated, and zero values are substituted to prepare the data for the prediction techniques. Subsequently, the output data is used to train and test the prediction-learning techniques. The training and testing portions of the dataset are separated to evaluate the learning models. Finally, the prediction output is obtained once the decision functions are executed, and a model evaluation is conducted. best models.

IV. RESULTS AND DISCUSSION

This section describes how the proposed model was evaluated. The evaluation uses the publicly available Google cluster workload trace dataset [5, 21]. The dataset comprises multiple concurrent activity traces for a month in a single 12K machine cluster. It includes traces of all requests and actions made by the cluster scheduler, resource utilization for each task over time, and traces of machine availability. Each trace describes several user-submitted jobs, and every job has one to ten tasks, which are programs to be run on an available machine. These tasks are typically carried out simultaneously rather than being gang-scheduled. Each task is given several specifications, such as priority, resource request (estimated maximum RAM and CPU needed), and occasionally restrictions (such as not operating on a machine without an external IP address). The range of these parameters is broader than that of conventional cluster workloads.

The trace also records every time a task is submitted, assigned to a machine, or rescheduled; this information enables users to look at the tasks, jobs, and scheduler behaviors. The trace also includes each machine's per-task resource utilization data. The trace lacks accurate information regarding the machine configuration and the purpose of the jobs. Although the trace does contain IDs for jobs, usernames, machine platforms, and configurations, these identifiers have been obscured by the trace providers, so users can only distinguish distributions of jobs, usernames, and machine characteristics throughout the trace. The trace contains six separate tables: task usage, task constraints, job events, machine events, task events, and machine attributes.

Each job in the dataset consists of multiple concurrent tasks that run on different machines. The parameters in the dataset include CPU utilization, memory usage, disk utilization, and so on. According to earlier research [5], less than 2% of jobs take more than one day. For evaluation, a long-running job (ID 1617658948) consisting of 60,171 tasks was selected. The job spans 20 days, with the first 15 days used for training and the remaining data for testing. The assessment includes univariate input where CPU, memory, disk, or disk I/O time is considered, and multivariate input where CPU, memory, disk, and disk I/O time inputs are considered.

Relevant data is collected, a predictive model is built, and the expected error is estimated. Various machine-learning algorithms are employed to determine the best model for predicting future resource usage, such as CPU, memory, disk utilization, and disk I/O time. A neural network with Adam and SGD optimizers, along with other machine learning techniques, is utilized to compare their Mean Absolute Error (MAE). compare their Mean Absolute Error (MAE), R-squared, root mean square error, and mean absolute percentage error.

Experimental Setup and Result Analysis

Python, which includes many libraries, simulates network traffic patterns. For the benefit of experimentation, the dataset is split into two sections. The first portion of the data was used to train the system, and the second portion was used to evaluate how accurate the forecast was. The Jupiter Notebook IDE has also been utilized. The add () function is used to add LSTM and dense layers to apply the LSTM model. Adam and SGD optimizers are employed to adjust the learning rate. The loss function employed with these optimizers are calculated using MSE. Other machine learning algorithms, such as MLP Regression, Random Forest, Decision Tree Regression, and Support Vector Regression, are employed on the dataset.

Four experiments are executed to predict CPU, memory, disk resources, and disk I/O time for different time series interval periods: the first one is for 3 minutes, the second is for 5 minutes, the third is for 8 minutes, and the last one is for the 10-minute time series period. models.

1) Three minutes_based on Google cluster workload experiment

Tables II and III display the comparative results among different machine learning techniques for predicting CPU, memory, disk and disk I/O time based on a 3-minute time series in terms of MAE, RMSE, R-squared score and MAPE for the univariate input case.

A. For the univariate input case

1) CPU:
   The strongest model overall is the Neural network with SGD optimizer (NN (SGD)), with the lowest RMSE (0.0226), the highest R-squared score (0.9993), and a competitive MAPE (0.0109).
2) Memory:
   Decision tree regression (DTR) is the best-performing model for predicting memory resources with the lowest RMSE (0.0059), highest R-squared (0.999), and competitive MAPE (0.0296).
3) Disk:
   Neural network with Adam optimizer (NN (Adam)) is the best-performing model with the lowest RMSE (0.0032), the highest R-squared (0.99999), and competitive MAPE (0.0026).
4) Disk I/O time:
   Neural network with Adam optimizer is the best-performing model with the lowest RMSE (0.0258), the lowest MAPE (0.006445), and competitive R-squared (0.9997).

Fig. 2-5 show graph plots of resource prediction based on the multivariate input case using different models.

TABLE II
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR UNIVARIATE INPUT CASE (3 MINUTES)

Resource	CPU				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
SVR	0.4988	0.1560	0.9691	0.0968	0.308305	0.2351	0.9480	0.0042
MLP	0.5078	0.1731	0.9620	0.6792	0.307782	0.1666	0.9739	0.0428
NN : Adam	0.5140	0.0342	0.9985	0.0105	0.32489	0.2391	0.9462	0.0094
NN (SGD)	0.5197	0.0226	0.9993	0.0109	0.305653	0.2396	0.9460	0.0011
RF	0.6952	0.0761	0.9926	1.0572	0.402572	0.4086	0.8430	1.6080
DTR	0.7135	0.0259	0.9991	0.2403	0.403575	0.0059	0.9999	0.0296
Proposed Model	0.4988 (SVR)	0.0226 (NN (SGD)	0.9993 (NN SGD)	0.0105 (NN : Adam)	0.305653 NN (SGD)	0.0059 DTR	0.9999 DTR	0.0011 NN (SGD)

TABLE III
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING DISK AND DISK I/O TIME FOR UNIVARIATE INPUT CASE (3 MINUTES)

Resource	Disk				Disk I/O Time			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
SVR	0.086	0.0095	0.999909	0.0080	0.41497	0.1758	0.9666	0.095272
MLP	0.0479	0.0125	0.999846	0.0130	0.45117	0.0636	0.9963	0.44703
NN : Adam	0.0447	0.0032	0.99999	0.0026	0.44729	0.0258	0.9967	0.006445
NN (SGD)	0.0766	0.0093	0.999914	0.0005	0.45117	0.0301	0.9904	0.008548
RF	0.0241	0.1074	0.988607	0.5479	0.55625	0.1542	0.9789	0.481701
DTR	0.025	0.0100	0.9999	0.0214	0.55276	0.0317	0.9997	0.201661
Proposed Model	0.0241 (RF)	0.0032 NN : Adam	0.99999 NN : Adam	0.0005 NN SGD	0.41497 SVR	0.0258 NN: Adam	0.9997 DTR	0.006445 NN: Adam

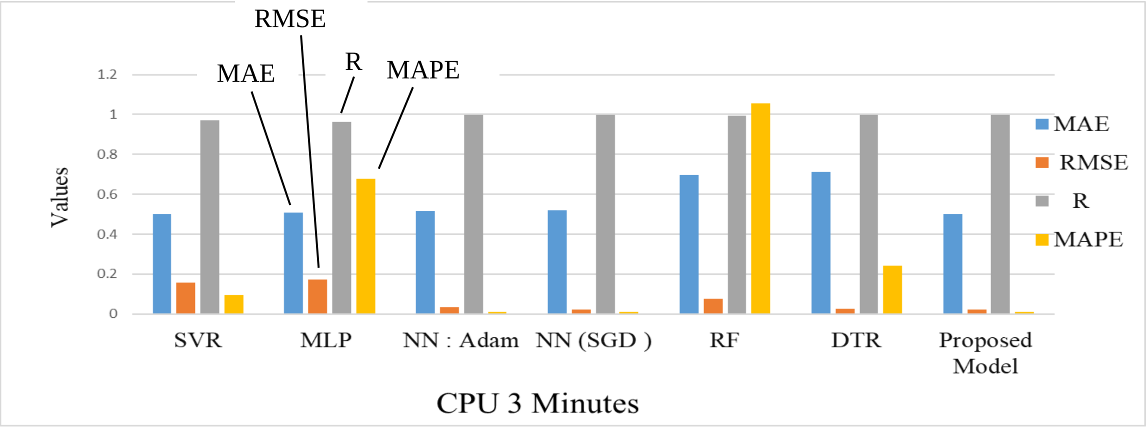

Fig. 2. Error rates of CPU prediction based on univariate input case (3 minutes)

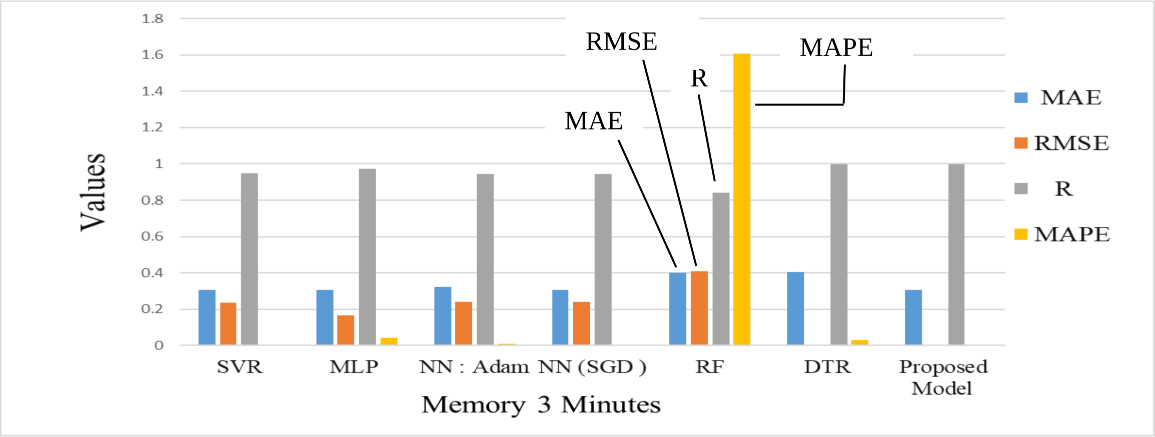

Fig. 3. Error rates of memory prediction based on univariate input case (3 minutes)

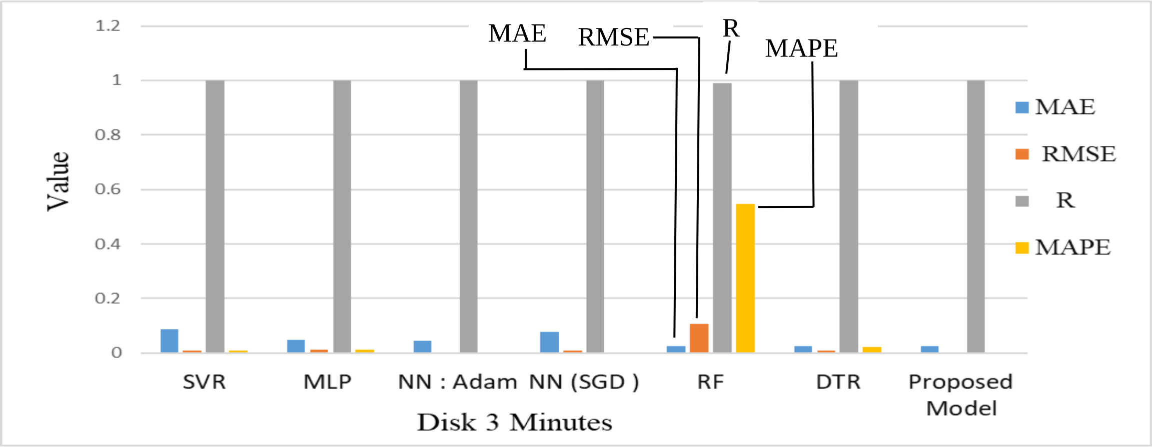

Fig. 4. Error rates of disk prediction based on univariate input case (3 minutes)

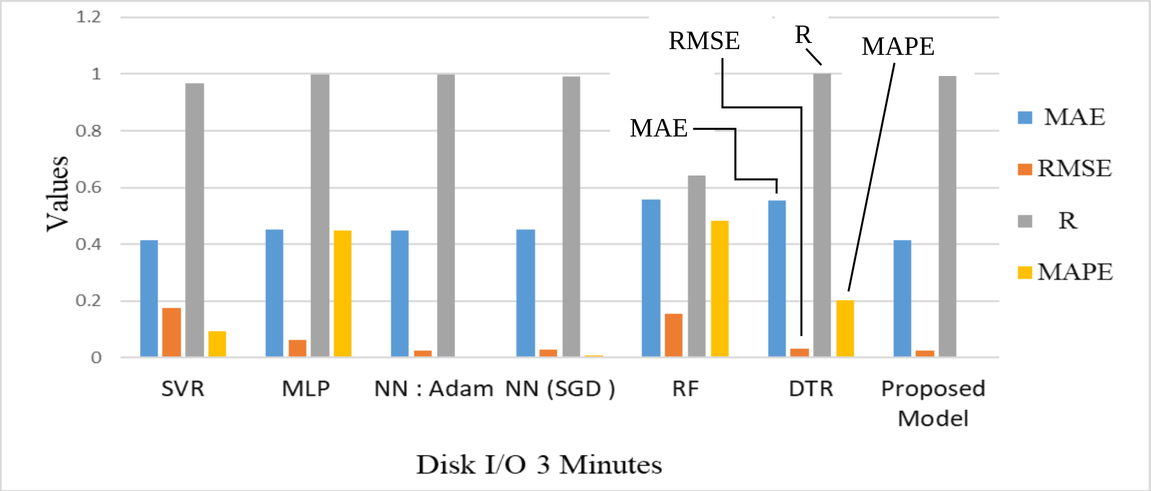

Fig. 5. Error rates of disk I/O time prediction based on univariate input case (3 minutes)

TABLE IV
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR MULTIVARIATE INPUT CASE (3 MINUTES)

	CPU				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.360444	0.221994	0.045	0.44421	0.221994	0.15090	0.967297	0.315929
NN (SGD)	0.361448	0.224403	0.046	0.44738	0.224403	0.08877	0.988682	0.217975
DTR	0.463975	0.283033	0.028	0.59682	0.283033	0.02923	0.998773	0.076925
RF	0.429661	0.256123	0.043	0.52869	0.256123	0.02443	0.999143	0.017358
SVR	0.360675	0.217427	0.071	0.42488	0.217427	0.13726	0.972939	0.214915
MLP	0.358721	0.255768	0.056	0.45901	0.255768	0.08046	0.990701	0.171582
Proposed Model	0.358721 (MLP)	0.217427 (SVR)	0.071 (SVR)	0.42488 (SVR)	0.217427 (SVR)	0.02443 (RF)	0.999143 (RF)	0.017358 (RF)

TABLE V
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING DISK RESOURCE AND DISK I/O TIME FOR MULTIVARIATE INPUT CASE (3 MINUTES)

Resource	Disk				Disk I/O			
Algorithms	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.0451	0.37622	0.827482	0.1976	0.44421	0.044426	0.997855	0.050749
NN (SGD)	0.0463	0.0808133	0.9920399	0.4524643	0.44738	0.078326	0.993333	0.152738
DTR	0.0275	0.2356643	0.9323079	0.0159632	0.59682	0.00235	0.999994	0.000869
RF	0.0425	0.2214388	0.9402335	0.0380015	0.52869	0.003246	0.999989	0.002503
SVR	0.071	0.5193225	0.671281	0.496123	0.42488	0.205646	0.954042	0.152219
MLP	0.0556	0.0779708	0.99259	0.4233347	0.45901	0.074005	0.994048	0.083783
Proposed Model	0.0275 (DTR)	0.080813 NN (SGD)	0.99259 (MLP)	0.015963 (DTR)	0.42488 (SVR)	0.00235 (DTR)	0.999994 (DTR)	0.000869 (DTR)

B. *For the multivariate input case*

Tables IV and V display the comparative results among different machine learning techniques for predicting CPU, memory, disk and disk I/O time based on a 3-minute time series in terms of MAE, RMSE, R-squared score and MAPE for the multivariate input case.

1) CPU:
The strongest model overall is support vector regression (SVR), which has the lowest RMSE (0.217427), the greatest R-squared (0.071), and the lowest MAPE (0.42488).

2) Memory:
Random forest(RF) is the best-performing model for predicting memory resources with the lowest RMSE (0.02443), the highest R-squared (0.999143), and the lowest MAPE (0.017358).

3) Disk*:*
Decision tree regression (DTR) is the best-performing model with the lowest MAE (0.0275), competitive R-Score (0.9323079), and the lowest MAPE (0.015963).

4) Disk I/O time*:*
Decision tree regression (DTR) is the strongest model with the lowest RMSE (0.00235), the lowest MAPE (0.000869), and the highest R-squared (0.999994).

Fig. 6-9 show graph plots of resource prediction based on the univariate input case using different models.

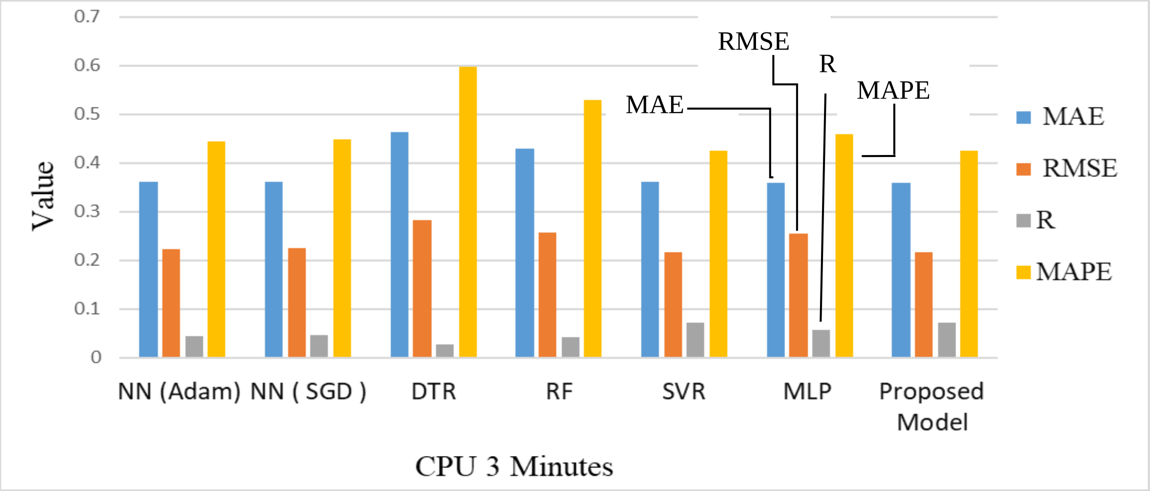

Fig. 6. Error rates of CPU prediction based on multivariate input case (3 minutes)

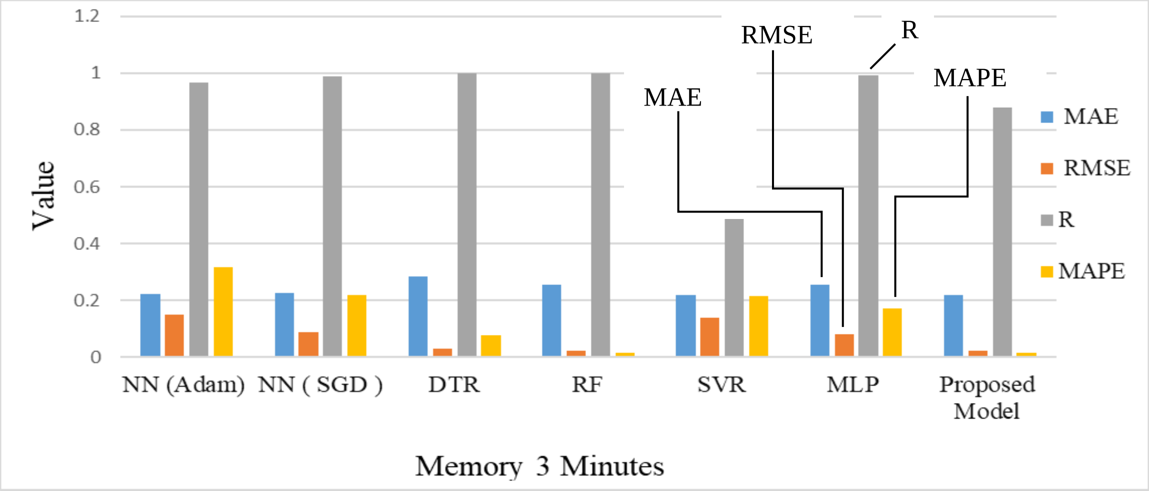

Fig. 7. Error rates of memory prediction based on multivariate input case (3 minutes)

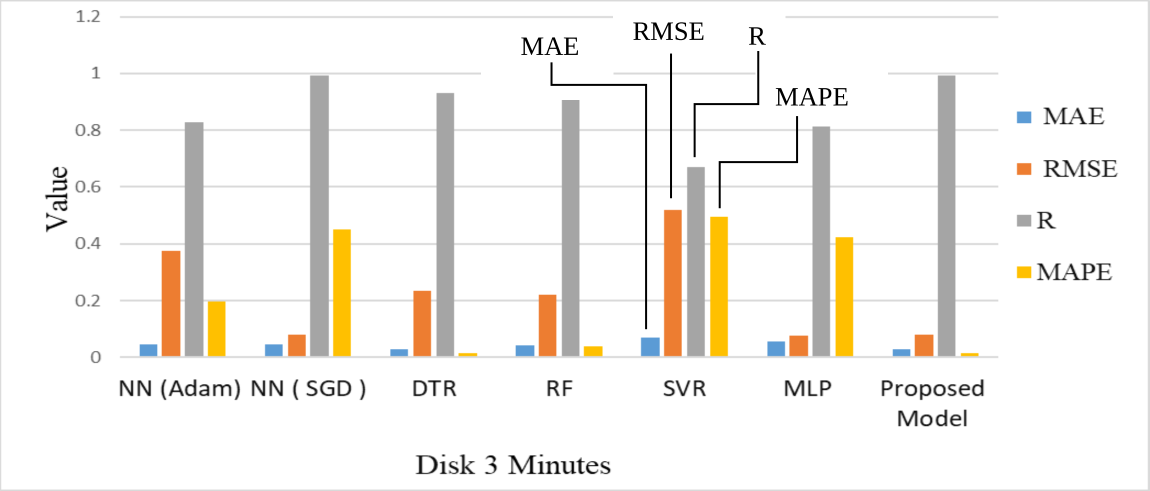

Fig. 8. Error rates of disk prediction based on multivariate input case (3 minutes)

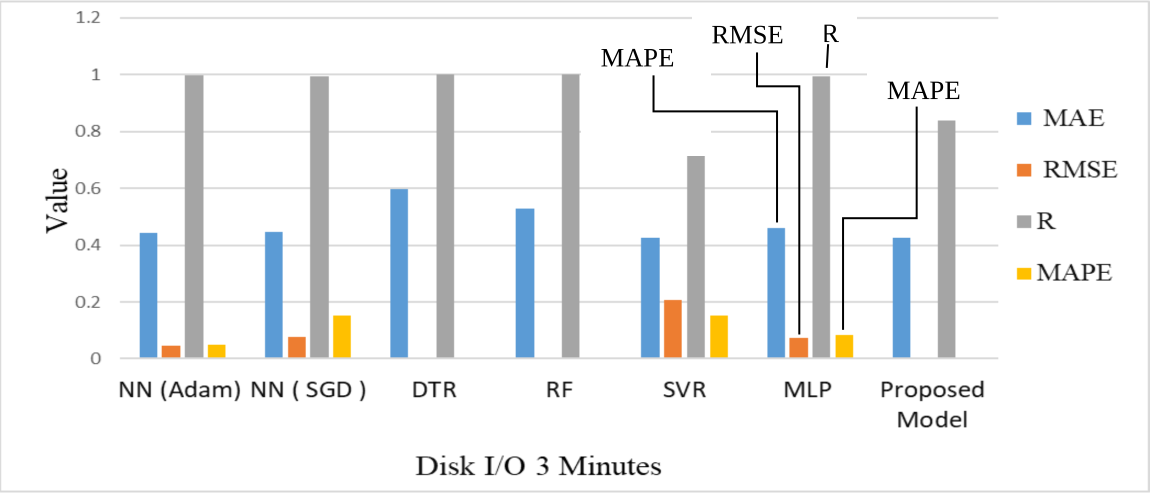

Fig. 9. Error rates of disk I/O time prediction based on multivariate input case (3 minutes)

TABLE VI
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR UNIVARIATE INPUT CASE (5 MINUTES)

Resource	CPU				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
RF	0.00309	0.122055	0.982564	0.466721	0.00535	0.049245	0.99733	0.867344
DTR	0.00451	0.078326	0.993333	0.152738	0.00825	0.037996	0.99841	0.218668
MLP	0.017509	0.00235	0.999994	0.000869	0.01868	0.016381	0.99970	0.008346
NN (SGD)	0.026268	0.003246	0.999989	0.002503	0.08413	0.009521	0.9999	0.021058
SVR	0.069424	0.205646	0.954042	0.152219	0.08007	0.211517	0.95077	4.093326
NN(Adam)	0.109813	0.074005	0.994048	0.083783	0.05313	0.007008	0.99994	0.478344
Proposed Model	0.00309 (RF)	0.00235 (MLP)	0.999994 (MLP)	0.000869 (MLP)	0.00535 (RF)	0.007008 NN(Adam)	0.99994 NN(Adam)	0.008346 (MLP)

TABLE VII
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING DISK AND DISK I/O FOR MULTIVARIATE INPUT CASE (5 MINUTES)

Resource	Disk				Disk I/O			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
RF	0.0081	0.007977	0.999937	0.003232	0.50269	0.123255	0.985174	0.032406
DTR	0.0086	0.013074	0.99983	0.009102	0.49772	0.035593	0.998764	0.032714
MLP	0.0245	0.013838	0.99981	0.001915	0.45689	0.044135	0.998099	0.004965
NN (SGD)	0.0673	0.013838	0.99981	0.00093	0.46779	0.054158	0.997138	0.002339
SVR	0.0445	0.078509	0.993884	0.229484	0.39977	0.301603	0.911226	0.519734
NN(Adam)	0.1305	0.018364	0.999665	0.030589	0.45841	0.015403	0.999768	0.116477
Proposed Model	0.0081 (RF)	0.007977 (RF)	0.999937 (RF)	0.00093 NN(SGD)	0.39977 (SVR)	0.015403 NN(Adam)	0.999768 NN(Adam)	0.002339 NN (SGD)

2) Five minutes_based on Google cluster workload experiment

Tables VI and VII show the comparative results between the different machine learning techniques for predicting CPU, memory, disk resources and disk I/O time based on a 5-minute time series in terms of MAE, RMSE, R-squared and MAPE for univariate input case. Based on the overall performance, the prediction algorithm is selected.

A. For the univariate input case

1) CPU:
The strongest model overall is MLP, with the lowest RMSE (0.00235), the highest R-squared (0.999994), and the lowest MAPE (0.999994).

2) Memory*:*
Neural network with Adam optimizer is the best model for predicting memory resources with the lowest RMSE (0.007008), the highest R-squared (0.99994), and a competitive MAPE.

3) Disk:
The random forest is the best model with the lowest MAE (0.0081), the lowest RMSE (0.007977), and the highest R-squared (0.999937).

4) Disk I/O:
- Lowest MAE: SVR (0.39977)
- Lowest RMSE: NN (Adam) 0.015403
- Highest R-squared: NN (Adam) 0.999768
- Lowest MAPE: NN (SGD) 0.002339

The neural network with Adam optimizer is the best performing model with the lowest RMSE (0.015403), the highest R-squared (0.999768), and a competitive MAPE.

Fig. 10-13 show graph plots of CPU, memory, disk utilization and disk I/O time prediction based on univariate input case using different models.

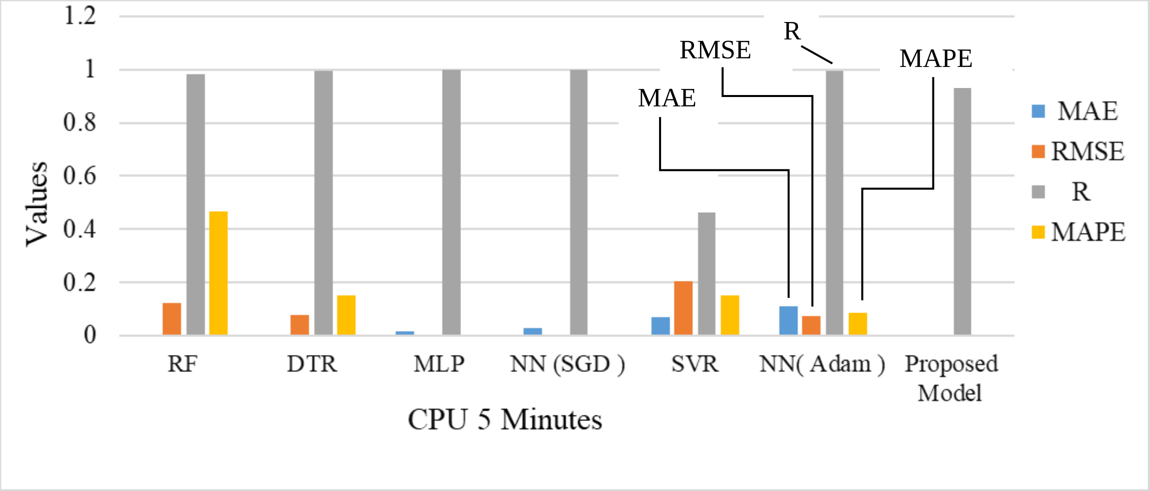

Fig. 10. Error rates of CPU prediction based on univariate input case (5 minutes)

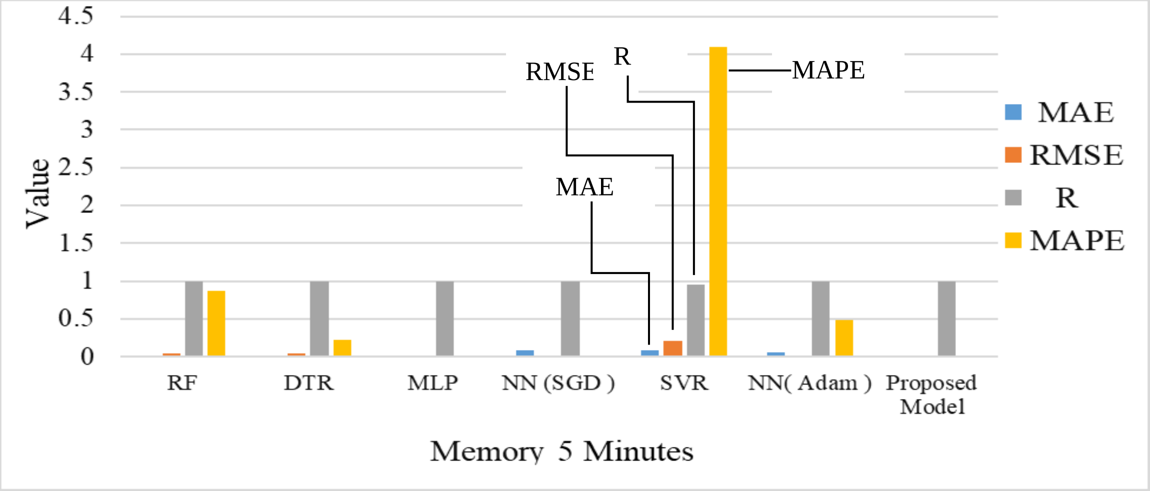

Fig. 11. Error rates of memory prediction based on univariate input case (5 minutes)

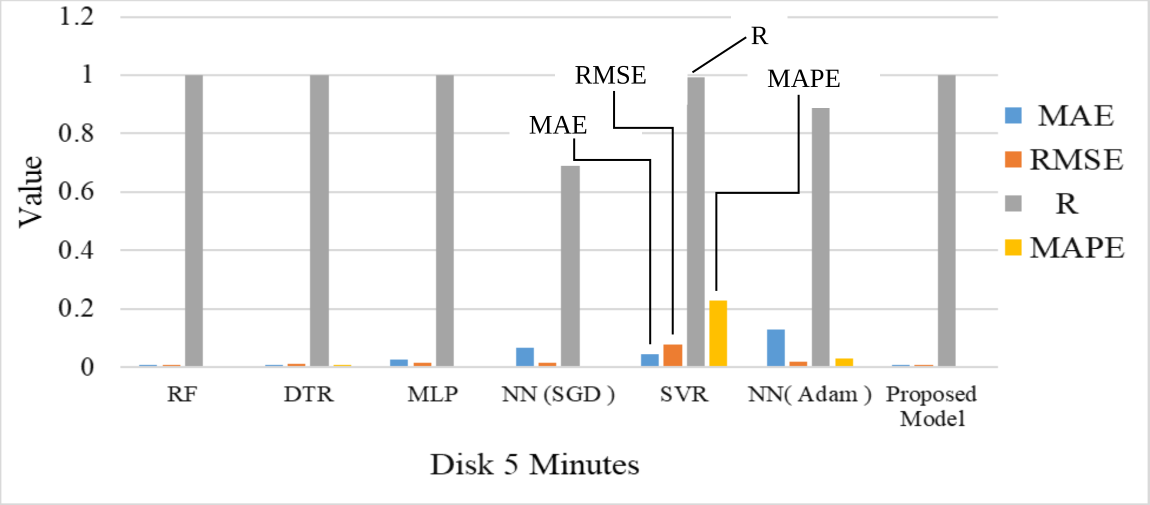

Fig. 12. Error rates of disk prediction based on univariate input case (5 minutes)

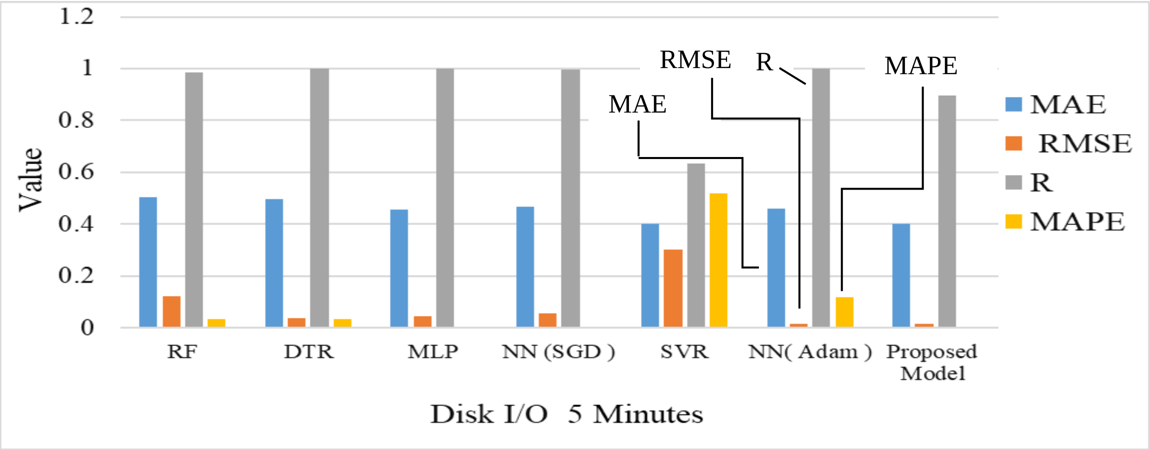

Fig. 13. Error rates of disk I/O time prediction based on univariate input case (5 minutes)

TABLE VIII
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR MULTIVARIATE INPUT CASE (5 MINUTES)

Resource	CPU				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.235311	0.097665	0.98767	0.106782	0.04321	0.199551	0.957439	0.086806
NN (SGD)	0.22859	0.041905	0.99773	0.172948	0.06609	0.046944	0.997645	0.449672
DTR	0.263386	0.027375	0.999031	0.010729	0.01029	0.057711	0.99644	0.018602
RF	0.243911	0.024709	0.999211	0.008131	0.00762	0.036377	0.998586	0.030925
SVR	0.231665	0.173	0.961313	0.213262	0.09189	0.394028	0.834057	0.831601
MLP	0.231642	0.04797	0.997026	0.181839	0.02297	0.045803	0.997758	0.474013
Proposed	0.22859	0.024709	0.999211	0.008131	0.00762	0.036377	0.998586	0.018602
Model	NN (SGD)	(RF)	(RF)	(RF)	(RF)	(RF)	(RF)	(DTR)

TABLE IX
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR MULTIVARIATE INPUT CASE (5 MINUTES)

Resource	Disk				Disk I/O			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.0299	0.173321	0.967892	0.265554	0.3039	0.043332	0.99828	0.042969
NN (SGD)	0.0321	0.130379	0.981832	0.458753	0.32166	0.039729	0.998554	0.062237
DTR	0.0089	0.057711	0.99644	0.018602	0.31994	0.007253	0.999952	0.008817
RF	0.0079	0.036377	0.998586	0.030925	0.31422	0.008169	0.999939	0.005852
SVR	0.0625	0.394028	0.834057	0.831601	0.30475	0.147615	0.980037	0.179233
MLP	0.0484	0.045803	0.997758	0.474013	0.31261	0.029243	0.999217	0.059648
Proposed	0.0079	0.036377	0.998586	0.018602	0.3039	0.007253	0.999952	0.005852
Model	(RF)	(RF)	(RF)	(DTR)	NN (Adam)	(DTR)	(DTR)	(RF)

B. For the multivariate input case (5 minutes)

Tables VIII and IX show the comparative results between the different machine learning techniques for predicting CPU, memory, disk resources and disk I/O time based on a 5-minute time series in terms of MAE, RMSE, R-squared and MAPE for univariate input case.

Based on the overall performance, the prediction algorithm is selected.

1) CPU:
   The strongest model overall is random forest (RF), with the lowest RMSE (0.024709), the highest R-squared (0.999211), and the lowest MAPE (0.008131).
2) Memory:
   The strongest model overall is random forest (RF), with the lowest MAE (0.00762), the highest R-squared (0.998586), and the lowest RMSE (0.998586).
3) Disk:
   The random forest is the best model with the lowest MAE (0.0081), the lowest RMSE (0.007977), and the highest R-squared (0.999937).
4) Disk I/O:
   - Lowest MAE: NN(Adam) (0.3039)
   - Lowest RMSE: DTR (0.007253)
   - Highest R-squared: DTR (0.99952)
   - Lowest MAPE: RF (0.005852)

   The decision tree regression(DTR) is the best performing model with the lowest RMSE (0.015403), the highest R-squared (0.999768), and a competitive MAPE.

Fig. 14-17 show graph plots of CPU, memory, disk utilization and disk I/O time prediction based on the multivariate input case using different models.

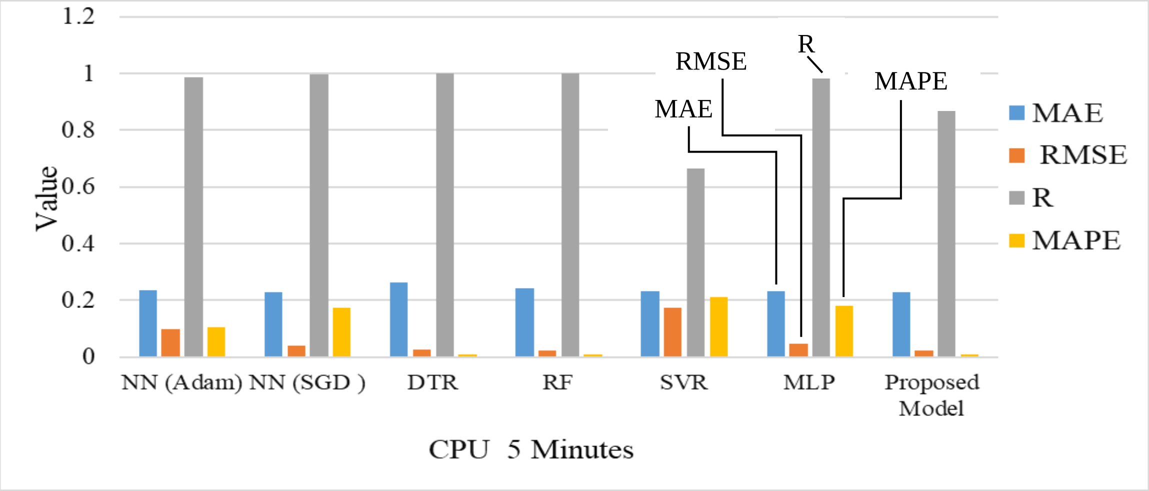

Fig. 14. Error rates of CPU prediction based on multivariate input case (5 minutes)

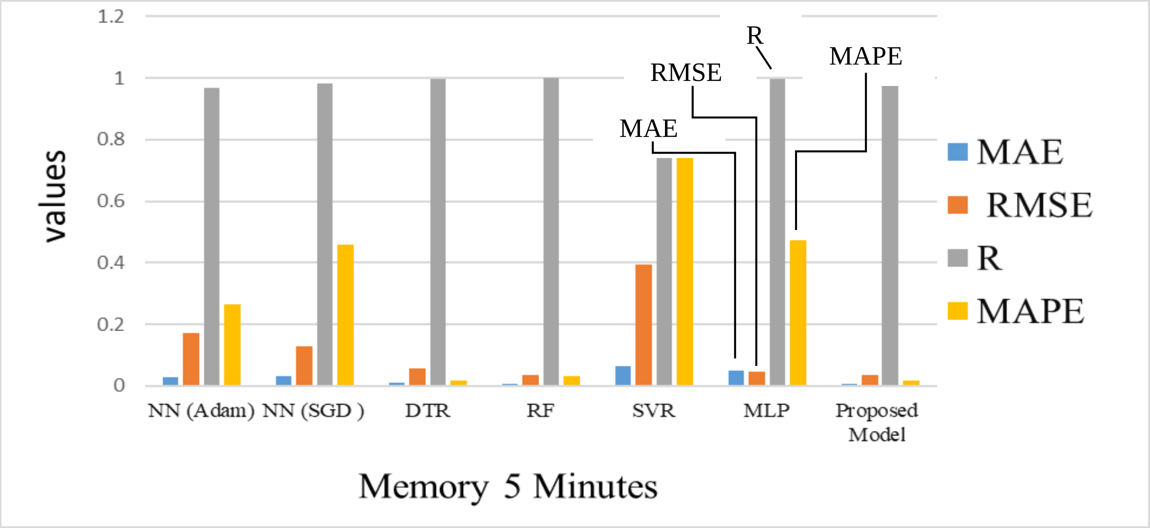

Fig. 15. Error rates of memory prediction based on multivariate input case (5 minutes)

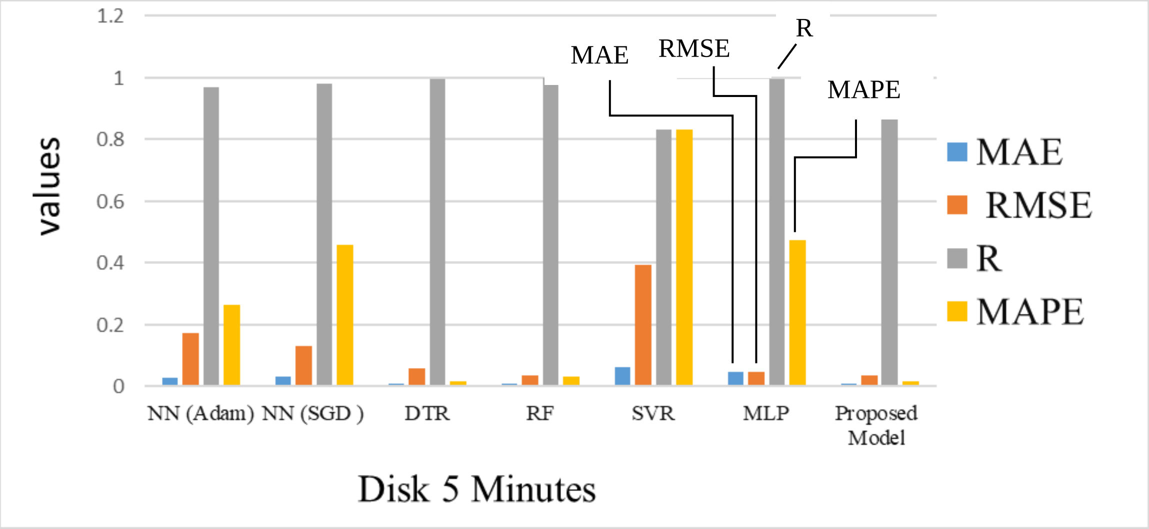

Fig. 16. Error rates of disk prediction based on multivariate input case (5 minutes)

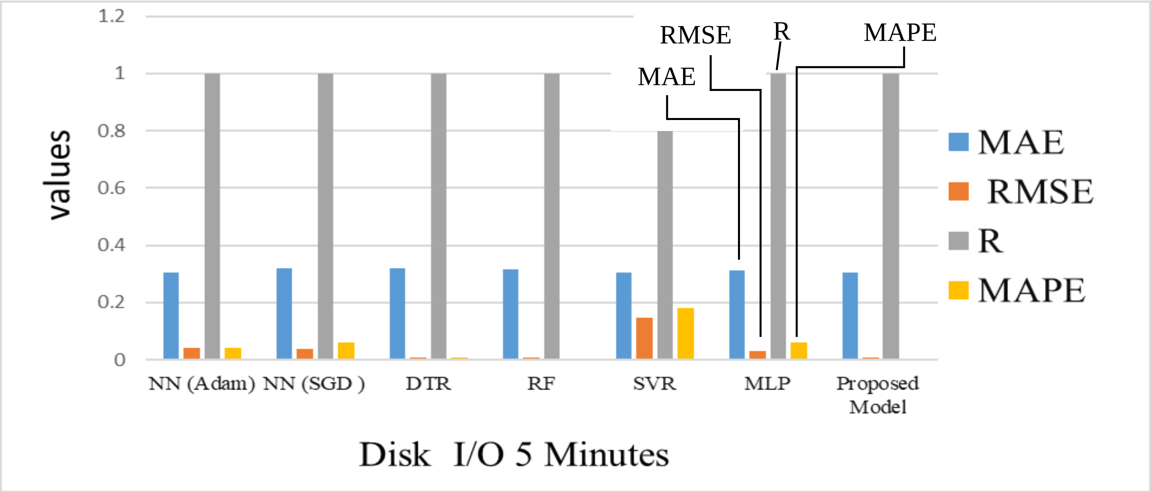

Fig. 17. Error rates of disk I/O time prediction based on multivariate input case (5 minutes)

TABLE X

THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR UNIVARIATE INPUT CASE (8 MINUTES)

Resource	CPU				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
SVR	0.535622	0.238342	0.939439	0.096071	0.35614	0.238342	0.939439	0.096071
MLP	0.542574	0.046193	0.997725	0.116116	0.37243	0.046193	0.997725	0.116116
NN (SGD)	0.549128	0.044148	0.997922	0.011884	0.38057	0.044148	0.997922	0.011884
NN (Adam)	0.554021	0.029066	0.999099	0.002089	0.40037	0.029066	0.999099	0.002089
RF	0.67532	0.465439	0.769051	0.932511	0.42949	0.465439	0.769051	0.932511
DTR	0.69774	0.010312	0.999887	0.129982	0.4323	0.010312	0.999887	0.129982
Proposed	0.535622	0.010312	0.999887	0.002089	0.35614	0.010312	0.999887	0.002089
Model	(SVR)	(DTR)	(DTR)	NN(Adam)	(SVR)	(DTR)	(DTR)	NN (Adam)

TABLE XI

THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR UNIVARIATE INPUT CASE (8 MINUTES)

Resource	Disk				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.0317	0.046439	0.997963	0.081657	0.44136	0.067981	0.992395	0.007013
NN (SGD)	0.0262	0.034314	0.998888	0.052533	0.50452	0.254914	0.893072	0.315054
DTR	0.0522	0.010191	0.999902	0.008867	0.51831	0.018276	0.99945	0.003415
RF	0.0474	0.011405	0.999877	0.004914	0.52695	0.027631	0.998744	0.001405
SVR	0.02346	0.164429	0.974467	0.518256	0.56166	0.185314	0.943491	0.707608
MLP	0.02119	0.022723	0.999512	0.078253	0.56175	0.010178	0.99983	0.083034
Proposed	0.02119	0.010191	0.999902	0.004914	0.44136	0.010178	0.99983	0.001405
Model	(MLP)	(DTR)	(DTR)	(RF)	NN (Adam)	(MLP)	(MLP)	(RF)

3) Eight minutes_based on Google cluster workload experiment

Tables X and XI show the comparative results between the different machine learning techniques for predicting CPU, memory, disk and disk I/O time based on a 3-minute time series in terms of MAE, RMSE, R-squared and MAPE for univariate and multivariate input cases. According to the overall performance, the prediction algorithm is selected.

A. For the univariate input case

1) CPU:

- Lowest MAE: SVR (0.535622)
- Lowest RMSE: DTR (0.010312
- Highest R-Score: DTR (0.999887)

The strongest model overall is decision tree regression(DTR), with the lowest RMSE, the greatest R- squared, and a competitive MAPE.

2) Memory:

The strongest model overall is decision tree regression(DTR), with the lowest RMSE, the greatest R-squared score, and a competitive MAPE.

3) Disk:

The strongest model overall is decision tree regression(DTR), with the lowest RMSE, the greatest R-squared score, and a competitive MAPE.

4) Disk I/O:

MLP is the best performing model due to the lowest RMSE, the highest R-squared score, and a competitive MAPE.

Fig. 18-21 show graph plots of CPU, memory, disk utilization and disk I/O time prediction based on univariate input cases using different models.

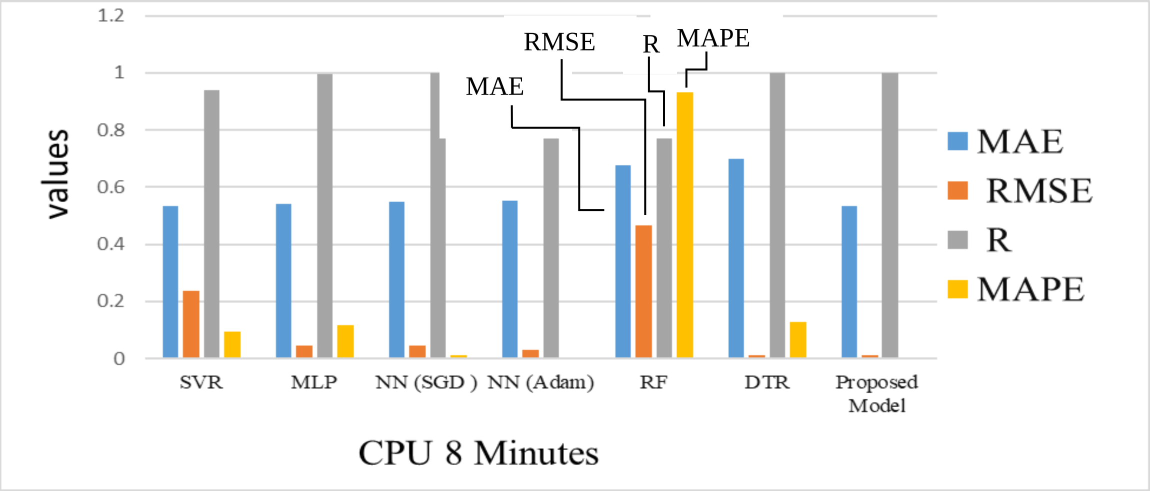

Fig. 18. Error rates of CPU prediction based on univariate input case (8 minutes)

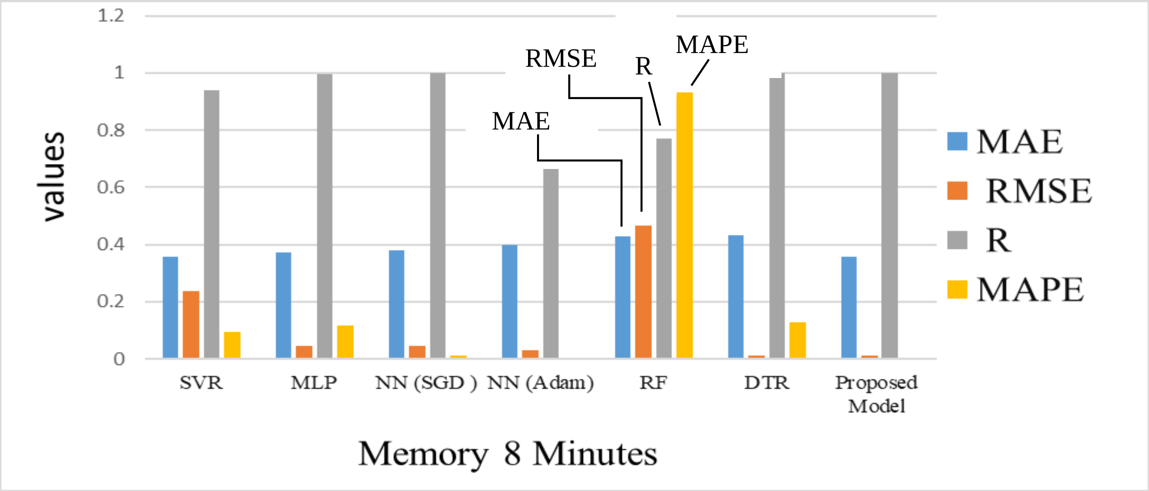

Fig. 19. Error rates of memory prediction based on univariate input case (8 minutes)

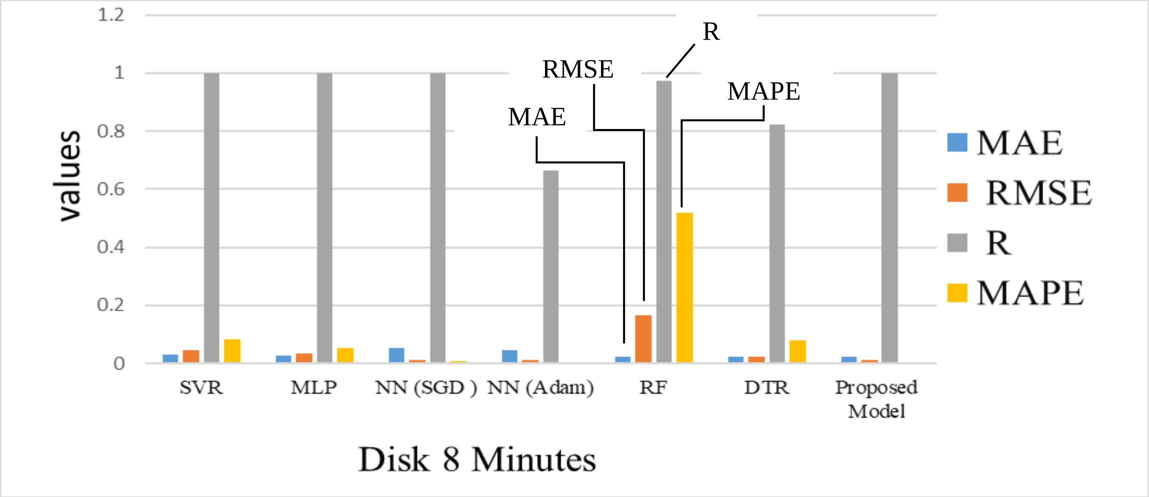

Fig. 20. Error rates of disk prediction based on univariate input case (8 minutes)

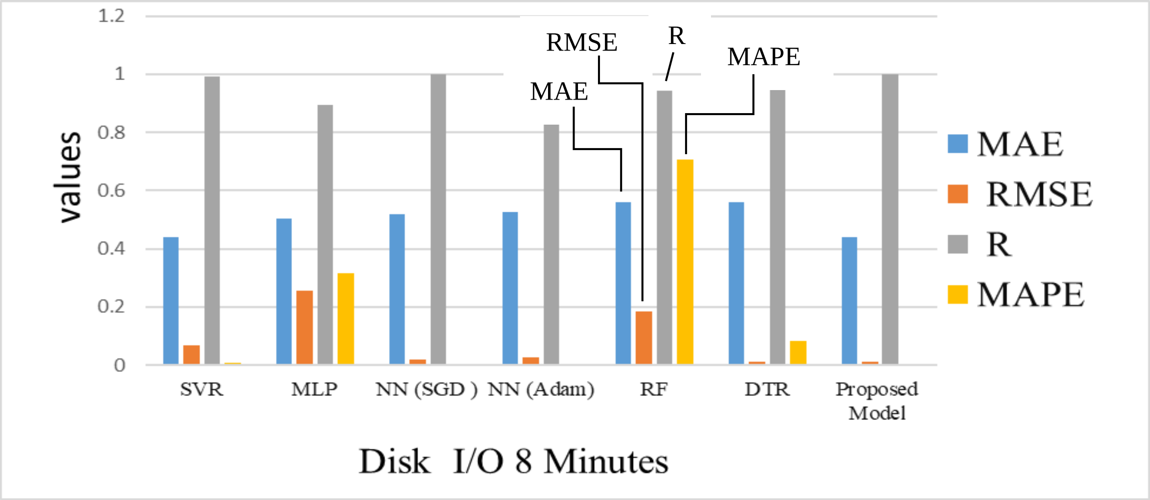

Fig. 21. Error rates of disk I/O time prediction based on univariate input case (8 minutes)

TABLE XII
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR UNIVARIATE INPUT CASE

Resource	CPU				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.521996	0.157255	0.945982	0.493233	0.36781	0.219875	0.916878	0.6089532
NN (SGD)	0.515038	0.110803	0.973181	0.40997	0.36598	0.142356	0.965157	0.78696414
DTR	0.627464	0.073786	0.988107	0.047354	0.45923	0.089605	0.986195	0.08595654
RF	0.557288	0.04578	0.995422	0.044496	0.41064	0.025485	0.998883	0.0422
SVR	0.507577	0.086698	0.983581	0.189912	0.36321	0.11188	0.978479	0.47506743
MLP	0.528993	0.075258	0.987628	0.227723	0.36425	0.083553	0.987997	0.5966089
Proposed Model	0.507577 (SVR)	0.04578 (RF)	0.995422 (RF)	0.044496 (RF)	0.36321 (SVR)	0.025485 (RF)	0.998883 (RF)	0.0422 (RF)

TABLE XIII
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR UNIVARIATE INPUT CASE

Resource	Disk				Disk I/O			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.03836	0.153322	0.897509	0.654282	0.41763	0.063856	0.99587602	0.06749
NN (SGD)	0.03896	0.108978	0.94822	0.9864893	0.4205	0.045306	0.997924	0.046412
DTR	0.02562	0.058454	0.985102	0.095833	0.52616	0.002901	0.9999914	0.002126
RF	0.02938	0.074308	0.975926	0.04858	0.45289	0.002968	0.99999109	0.002243
SVR	0.06274	0.08581	0.967896	0.42146	0.34021	0.073859	0.99448276	0.069671
MLP	0.05149	0.069764	0.97878	0.38023	0.4166	0.073579	0.9945244	0.05711
Proposed Model	0.02562 (DTR)	0.058454 (DTR)	0.985102 (DTR)	0.04858 (RF)	0.34021 (SVR)	0.002901 (DTR)	0.9999914 (DTR)	0.002126 (DTR)

B. For the multivariate input case

Tables XII and XIII show the comparative results between the different machine learning techniques for predicting CPU, memory, disk and disk I/O time based on a 3-minute time series in terms of MAE, RMSE, R-squared and MAPE for univariate and multivariate input cases.

Based on the overall performance, the prediction algorithm is selected.

1) CPU

The strongest model overall is support vector regression (SVR), which has the lowest RMSE (0.217427), greatest R-SCORE (0.071), and lowest MAPE (0.42488).

2) Memory

Random forest(RF) is the best performing models for predicting memory resources due to the lowest RMSE (0.02443), highest R-SCORE (0.999143), and lowest MAPE (0.017358).

3) Disk

- Lowest MAE: SVR (0.535622)
- Lowest RMSE: DTR (0.010312
- Highest R-Score: DTR (0.999887)
- Lowest MAPE:

Decision tree regression (DTR) is the best performing model due to the lowest MAE (0.0275), competitive R-Score (0.9323079), and lowest MAPE (0.015963).

4) Disk I/O time

- Lowest MAE: SVR (0.535622)
- Lowest RMSE: DTR (0.010312
- Highest R-Score: DTR (0.999887)

Decision tree regression (DTR) is the best performing model due to the lowest RMSE (0.00235), lowest MAPE (0.000869), and highest R-Score (0.999994).

Fig. 22-25 show graph plots of CPU, memory, disk utilization and disk I/O time prediction based on univariate input cases using different models.

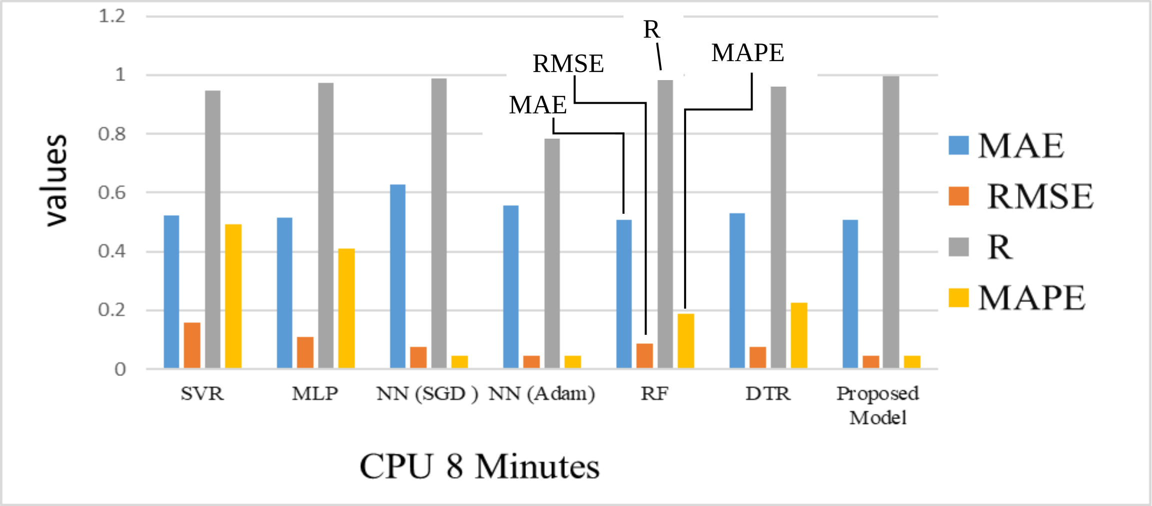

Fig. 22. Error rates of CPU prediction based on univariate input case (8 minutes)

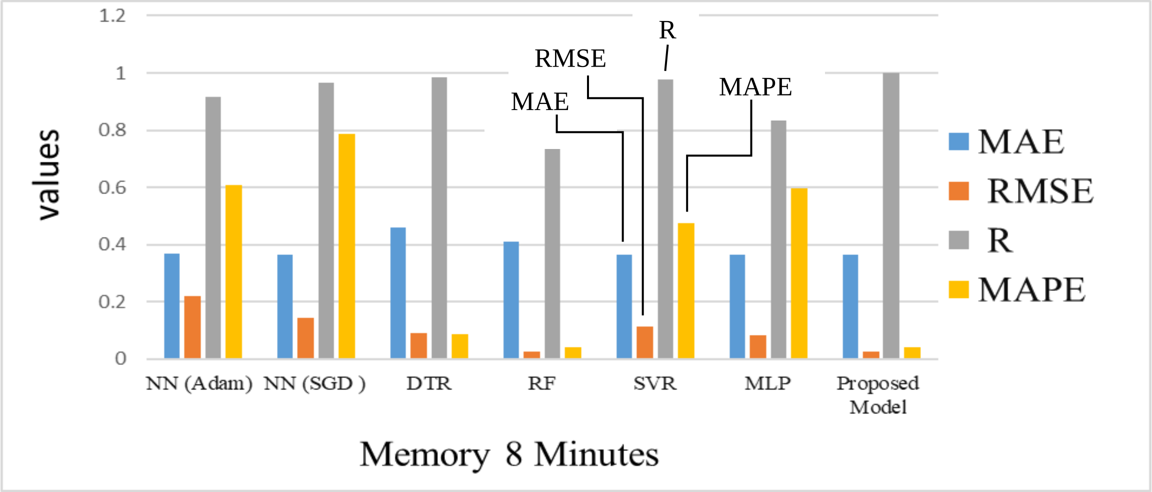

Fig. 23. Error rates of memory prediction based on univariate input case (8 minutes)

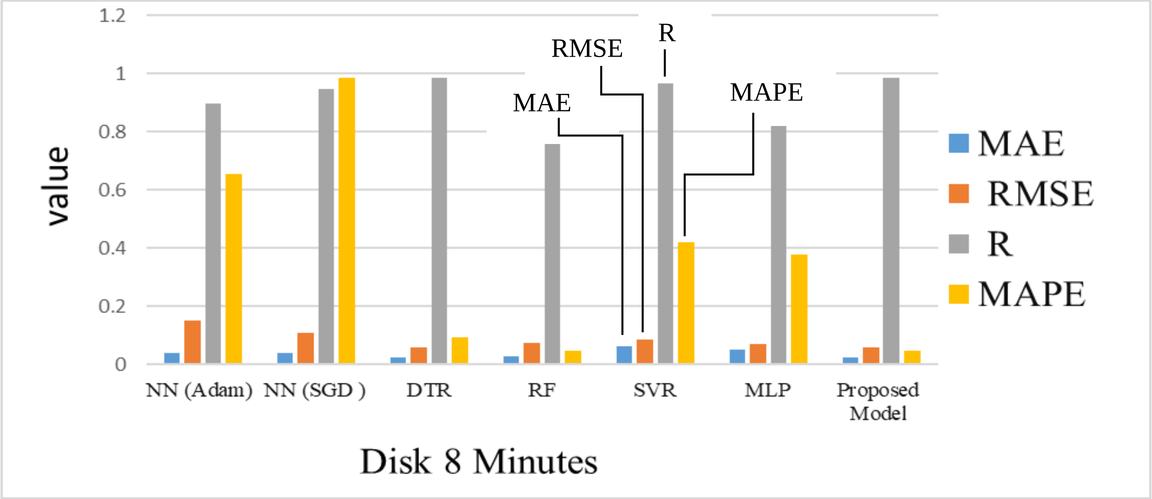

Fig. 24. Error rates of disk prediction based on univariate input case (8 minutes)

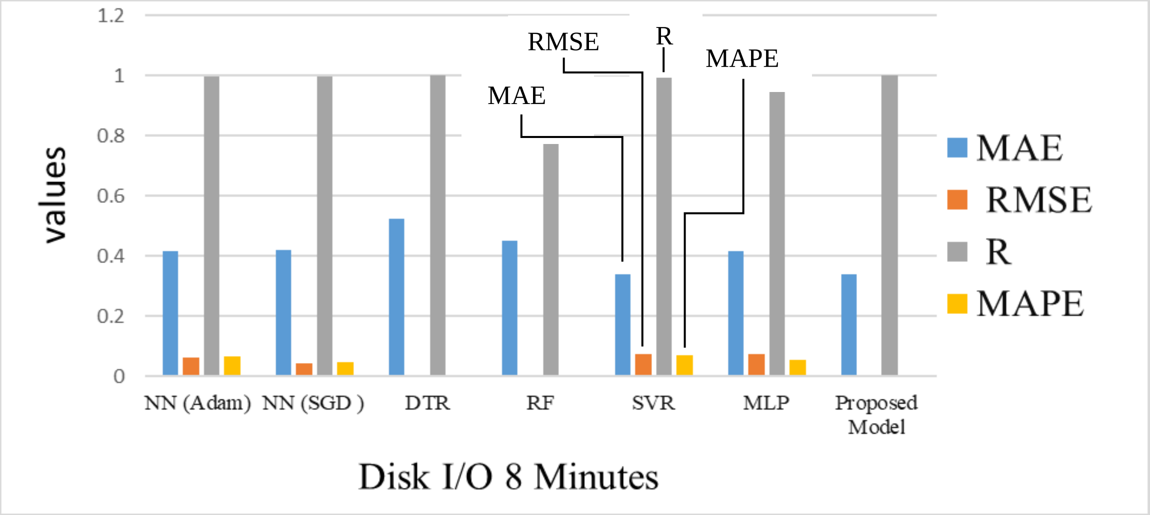

Fig. 25. Error rates of disk I/O time prediction based on univariate input case (8 minutes)

TABLE XIV
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR UNIVARIATE INPUT CASE (10 MINUTES)

Resource	CPU				Memory			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
SVR	0.474025	0.663891	0.74368	0.227878	0.41594	0.487619	0.813886	1.094633
NN (Adam)	0.516387	0.313786	0.94273	0.479215	0.42253	0.207093	0.96643	0.366268
MLP	0.529278	0.389173	0.91192	0.06905	0.41628	0.190544	0.971581	0.089674
NN (SGD)	0.547228	0.365375	0.92236	0.047003	0.45695	0.197291	0.969532	0.098654
DTR	0.612037	0.909082	0.51939	0.341817	0.4814	0.630585	0.688753	0.908327
RF	0.613433	0.143439	0.98803	0.197735	0.47225	0.087924	0.993948	0.473227
Proposed Model	0.474025 (SVR)	0.143439 (RF)	0.98803 (RF)	0.047003 NN (SGD)	0.41594 (SVR)	0.087924 (RF)	0.993948 (RF)	0.089674 (MLP)

TABLE XV
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING DISK AND DISK I/O TIME FOR UNIVARIATE INPUT CASE (10 MINUTES)

Resource	DISK				Disk I/O time			
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
SVR	0.02964	0.670013	0.663764	6.083757	0.38072	0.09423	0.991877	0.090429
NN (Adam)	0.05175	0.250217	0.953106	3.895048	0.45544	0.08901	0.992753	0.097126
MLP	0.02811	0.313985	0.926159	7.276436	0.44887	0.03318	0.998992	0.017226
NN (SGD)	0.11032	0.314305	0.926009	6.430044	0.4629	0.03502	0.998877	0.016795
DTR	0.0395	0.87389	0.428007	28.52868	0.52798	0.273451	0.931604	0.125950
RF	0.0364	0.065983	0.996739	0.20902	0.54809	0.104694	0.989974	0.070609
Proposed Model	0.02811 (MLP)	0.065983 (RF)	0.996739 (RF)	0.20902 (RF)	0.38072 (SVR)	0.03318 (MLP)	0.998992 (MLP)	0.016795 NN(SGD)

4) ten minutes_based on Google cluster workload experiment

Tables XIV and XV show the comparative results between the different techniques for predicting CPU, memory, disk and disk I/O time based on a 3-minute time series in terms of MAE, RMSE, R-squared and MAPE for univariate and multivariate input cases. According to the overall performance, the prediction algorithm is selected.

A. For the univariate input case

1) CPU:
   - Lowest MAE: SVR (0.474025)
   - Lowest RMSE: RF (0.143439)
   - Highest R-Score: RF (0.98803)

   The strongest model overall is random forest(RF), with the lowest RMSE and the highest R-squared.

2) Memory:
   The strongest model overall is random forest (RF), with the lowest RMSE, the highest R-squared, and a competitive MAPE.

3) Disk:
   The strongest model overall is random forest, with the lowest RMSE, the highest R-squared, and the lowest MAPE.

4) Disk I/O:
   MLP is the best performing model due to the lowest RMSE (0.03318), the highest R-squared (0.998992), and a competitive MAPE.

Fig. 26-29 show graph plots of CPU, memory, disk utilization and disk I/O time prediction based on univariate input cases using different models.

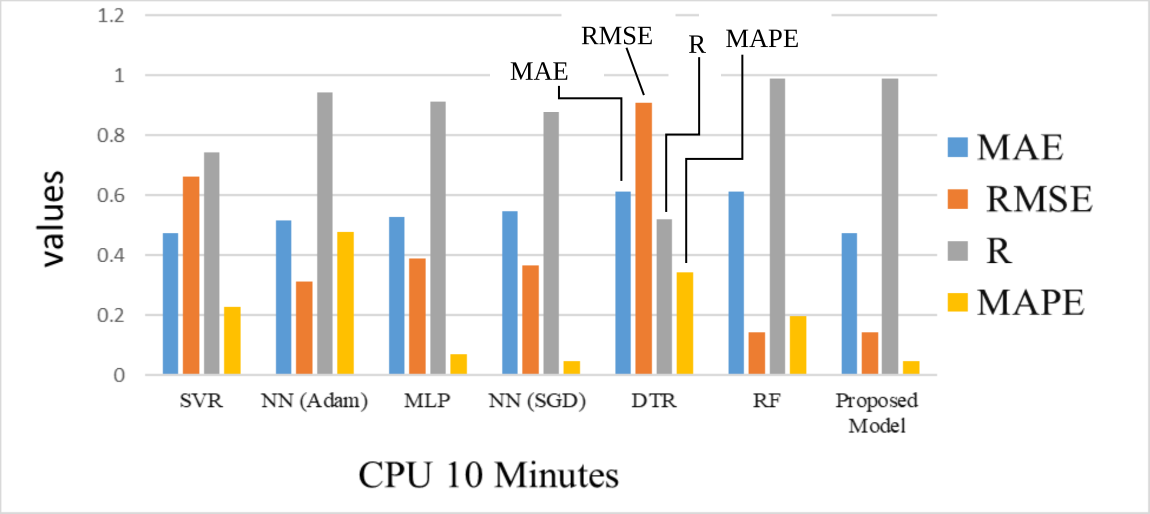

Fig. 26. Error rates of CPU prediction based on univariate input case (10 minutes)

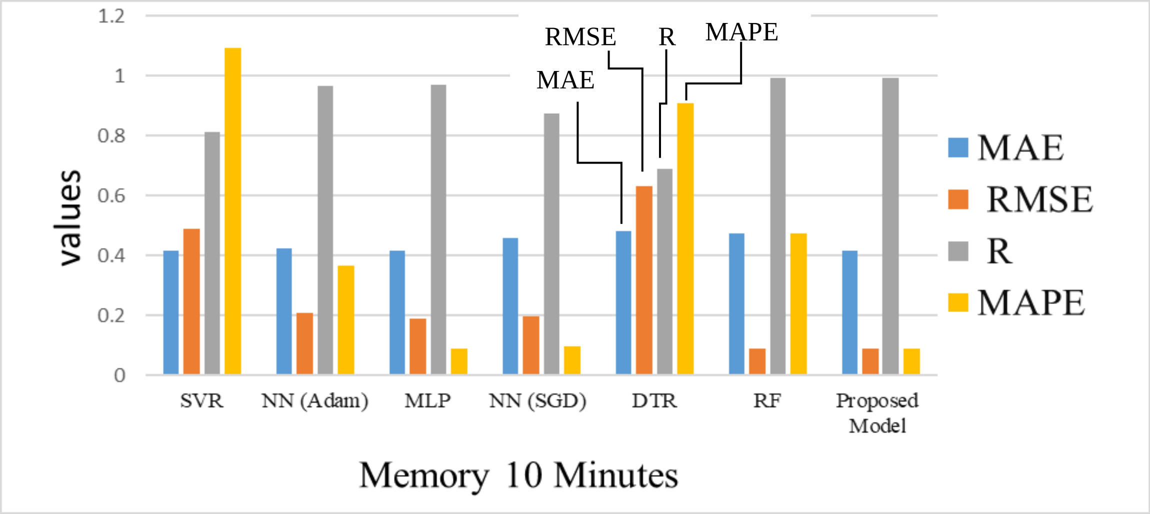

Fig. 27. Error rates of memory prediction based on univariate input case (10 minutes)

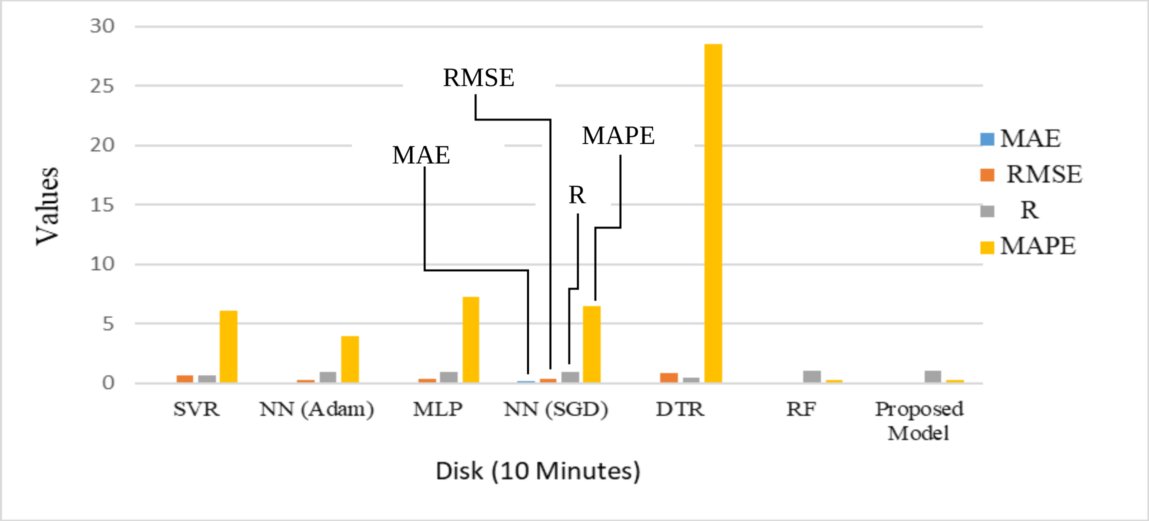

Fig. 28. Error rates of disk prediction based on univariate input case (10 minutes)

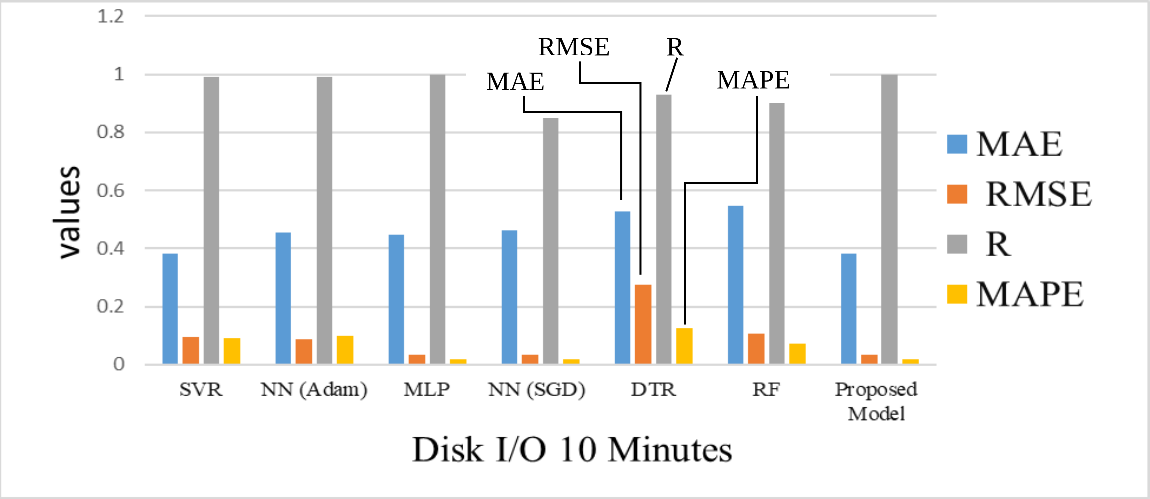

Fig. 29. Error rates of disk I/O time prediction based on univariate input case (10 minutes)

TABLE XVI
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING CPU AND MEMORY FOR MULTIVARIATE INPUT CASE

Resource		CPU				Memory		
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.58783	0.68781	0.724881	0.495091	0.41494	0.468779	0.82799	0.327559
NN (SGD)	0.58665	0.322915	0.93936	0.405493	0.41448	0.198958	0.969016	0.690024
DTR	0.927125	0.389173	0.911922	0.069059	0.55296	0.190544	0.971581	0.089674
RF	0.722019	0.000421	0.922364	0.017004	0.48984	0.197291	0.969533	0.098655
SVR	0.544754	0.909082	0.519392	0.341818	0.38989	0.630585	0.688753	0.908327
MLP	0.590027	0.365375	0.988035	0.197735	0.40387	0.087924	0.993949	0.473227
Proposed	0.544754	0.000421	0.988035	0.017004	0.38989	0.087924	0.993949	0.089674
Model	(SVR)	(RF)	(MLP)	(RF)	(SVR)	(MLP)	(MLP)	(DTR)

TABLE XVII
THE ERRORS RATES ACHIEVED BY ALL THE TECHNIQUES FOR PREDICTING DISK AND DISK I/O TIME FOR MULTIVARIATE INPUT CASE (10 MINUTES)

Resource		Disk				Disk I/O Time		
Algorithm	MAE	RMSE	R	MAPE	MAE	RMSE	R	MAPE
NN (Adam)	0.05958	0.715604	0.616450	5.10479054	0.44105	0.094505	0.991831	0.086884
NN (SGD)	0.06416	0.255068	0.951270	4.03063223	0.45475	0.075028	0.994851	0.091918
DTR	0.04681	0.313985	0.926159	0.20902026	0.4695	0.033188	0.998993	0.017226
RF	0.04145	0.314305	0.926009	6.4300449	0.45868	0.035027	0.998878	0.016796
SVR	0.0935	0.87389	0.428007	28.5286821	0.36888	0.273451	0.931604	0.125951
MLP	0.07062	0.065983	0.996739	7.27643695	0.44935	0.104694	0.989974	0.070609
Proposed	0.04145	0.065983	0.996739	0.20902026	0.36888	0.033188	0.998993	0.016796
Model	(RF)	(MLP)	(MLP)	(DTR)	(SVR)	(DTR)	(RF)	(RF)

B. For the multivariate input case

Tables XVI and XVII show the comparative results between the different machine learning techniques for predicting CPU, memory, disk and disk I/O time based on a 3-minute time series in terms of MAE, RMSE, R-squared and MAPE for the multivariate input cases.

Based on the overall performance, the prediction algorithm is selected.

1) CPU

The strongest model overall is MLP, which has the lowest RMSE (0.143439), the highest R-squared (0.988035), and a competitive MAPE.

2) Memory

MLP is the best performing models for predicting memory resources with the lowest RMSE (0.087924), highest R-squared (0.993949), and a competitive MAPE.

3) Disk

- Lowest MAE: RF (0.04145)
- Lowest RMSE: MLP (0.065983)
- Highest R-Score: MLP (0.996739)
- Lowest MAPE: DTR (0.20902026)

MLP is the best performing model with the lowest RMSE (0.065983), the highest R-squared (0.998993), and a competitive MAPE.

4) Disk I/O time

Decision tree regression (DTR) is the best performing model due to the lowest RMSE (0.00235), lowest MAPE (0.000869), and highest R-Score (0.999994).

Fig. 30-33 show graph plots of CPU, memory, disk utilization and disk I/O time prediction based on univariate input cases using different models.

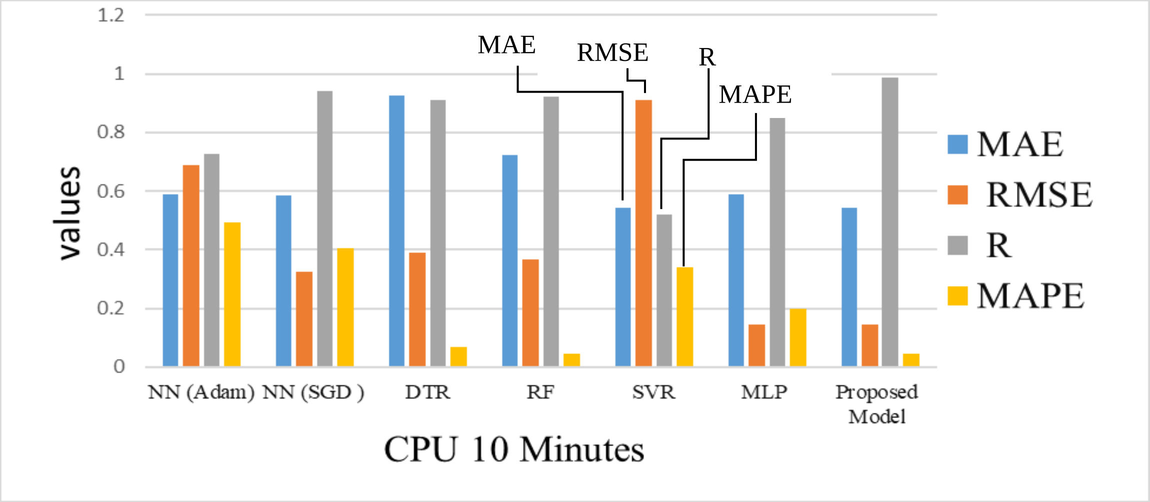

Fig. 30. Error rates of CPU prediction based on multivariate input case (10 minutes)

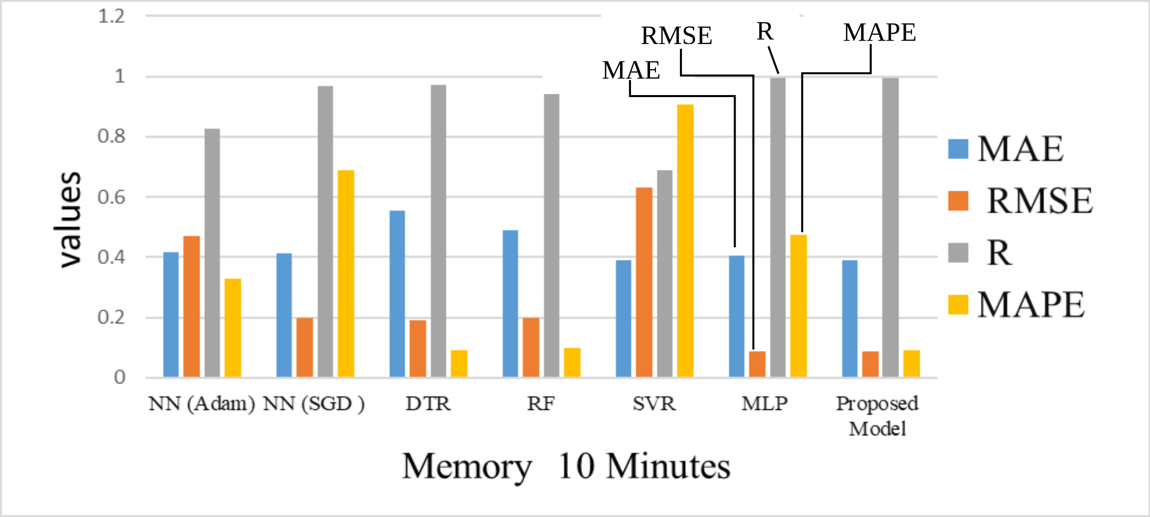

Fig. 31. Error rates of memory prediction based on multivariate input case (10 minutes)

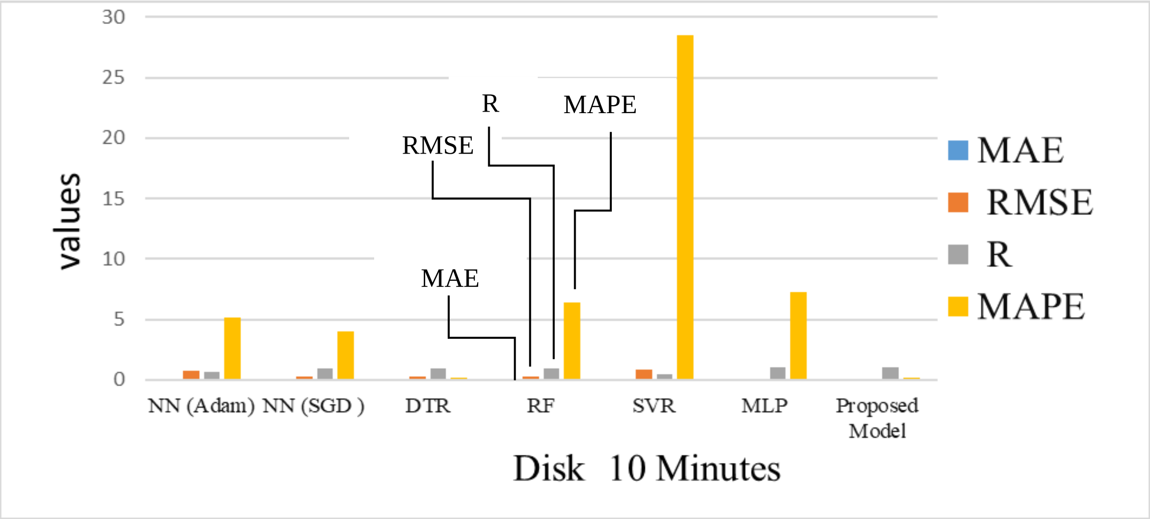

Fig. 32. Error rates of disk prediction based on multivariate input case (10 minutes)

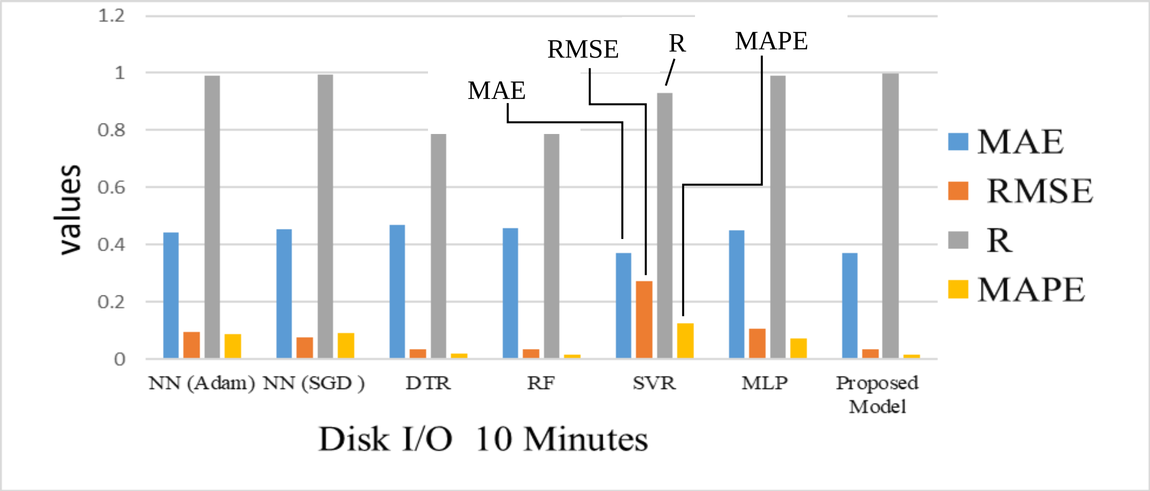

Fig. 33. Error rates of disk I/O time prediction based on multivariate input case (10 minutes)

TABLE XVIII
EVALUATION OF PREDICTION MODELS

Ref	Resources predicted	Dataset	Performance metric	time series		Data input case	Final results	
							Algorithm/Resource	Error Rates
[6]	Task priority	Google Cluster trace	MRPE	Not specified	SVM LR	Univariate	SVM	(MRPE) 0.5107
							LR	(MRPE) 0.7908
							Classified prediction	(MRPE) 0.4677
[9]	CPU	Google Cluster trace and Unix systems data	MSE	5 Minutes	LSTM	Univariate	LSTM	MSE (0.00045)
[17]	CPU	Google cluster trace and Planet Lab workload trace	RMSE MAPE	10 minutes	GD LM	Multivariate	GD	RMSE (0.0088)
								MAPE (0.0331)
							LM	RMSE (0.0085)
								MAPE (0.0321)
[26]	CPU Memory	Google cluster Trace	MAE	5 minutes	FLGAPSO	Univariate/ Multivariate	CPU (univariate) Memory (univariate) CPU (multivariate) Memory (multivariate)	MAE (0.25) MAE (0.018) MAE (0.33) MAE (0.026)
Proposed Model	CPU Memory Disk usage Disk I/O time	Google cluster data	MAE RMSE R-squared MAPE	3 minutes 5 minutes 8 minutes 10 minutes	RF DTR MLP SVR NN: SGD NN:Adam	Univariate/ Multivariate	5 (minutes) CPU (univariate) Memory (univariate) CPU (multivariate) Memory (multivariate)	 MAE (0.00309) (RF) MAE (0.00535) (RF) MAE (0.22859) (RF) MAE (0.00762) (RF)
							10 minutes CPU (multivariate)	 RMSE (0.000421) (RF) MAPE (0.017004) (RF)

V. RESULTS COMPARISON AND DISCUSSION

Table XVIII demonstrates the ways in which different datasets and algorithms have been applied to forecast different types of cloud resources.

In [6], their research focus on classification of the task priority based on the resources specifications of each task. Then based on the task category, the suitable prediction algorithm is assigned which is support vector machine or linear regression. Their model focuses on task classification

In [9] and [17], the authors focused on predicting a single resource, specifically CPU utilization, without including other resources such as memory and disk utilization in their model. They also worked on a single type of prediction input case, either univariate or multivariate, but not both: in [9], the authors focused on the univariate input case, while in [17], they focused on the multivariate input case.

In [26], the authors focused on predicting a wider range of resources, including CPU and memory resources. They worked on two types of input cases: univariate and multivariate input cases. However, their work was only evaluated using one performance metric, which is the MAE error, and they focused on a single time series interval (five minutes).

This paper utilizes various prediction algorithms to forecast a broader range of resources such as disk usage and disk I/O time. The predictions are made based on both univariate and multivariate input cases. The prediction is conducted over different time series intervals ranging from three to ten minutes. The work is evaluated by different performance metrics including MAE, MAPE, RMSE, and R-squared score.

Comparison of error rates

1) For CPU resources (5 Minutes):
   - In [9], the lowest MSE error achieved by LSTM for predicting CPU resources is 0.00045, with an equivalent RMSE of 0.0212. However, in the proposed model, using MLP performs better with a lower RMSE of 0.00235.
   - In [26], the lowest MAE error achieved by using the neural network with genetic algorithm and particle swarm optimization is 0.25, but in the proposed model, using random forest lowers the error to 0.00309.
2) For CPU resources (10 minutes):
   - In [17], the lowest errors of GD (Gradient Descent)
     - RMSE: 0.0088
     - MAPE: 0.0331
   - Levenberg-Marquardt (LM)
     - RMSE: 0.0085
     - MAPE: 0.0321
   - In the proposed model, using Random Forest (RF)
     - RMSE 0.000421
     - MAPE 0.017004

Using Random Forest (RF) significantly outperforms both GD and LM, achieving the lowest RMSE and MAPE.

3) For memory resources
   - In [26], the lowest error rates achieved by using FLGAPSO are:
     - Univariate input case: MAE (0.018)
     - Multivariate input case: MAE (0.026)
   - The proposed model, using random forest achieves lower error rates:
     - Univariate input case: MAE (0.00535)
     - Multivariate
     - MAE (0.00762)

Using random forest outperforms the other models for predicting memory resources with lower MAE error.

VI. Conclusion

The associate One of the most important challenges in managing uncertainty in cloud computing settings is predicting the utilization of cloud resources. Resources are allocated to user apps in cloud computing, which can be accessed via the Internet from any location. The resources need to be dynamically scaled to handle many users to optimize utilization, reduce energy consumption, and maintain cost-effectiveness while improving quality of service (QoS). This paper utilizes various prediction algorithms to forecast a broader range of resources such as CPU, memory, disk usage and disk I/O time. the algorithms used are neural networks (NN) with Adam and SGD optimizers, MLP regression, random forest, decision tree regression, and support vector regression. The predictions are made based on both univariate and multivariate input cases. The prediction is conducted over different time series intervals ranging from three to ten minutes. The work is evaluated by different performance metrics including MAE, MAPE, RMSE, and R-squared score.

The findings demonstrate that when compared to conventional methods, the proposed model produces outcomes with higher accuracy. It may also be inferred that the prediction of univariate and multivariate resource utilization is difficult due to the potential for abrupt and excessive changes in resource utilization. in resource utilization.

References

[1] Muteeh, Arfa, Muhammad Sardaraz, and Muhammad Tahir, "MrLBA: Multi-Resource Load Balancing Algorithm for Cloud Computing using Ant Colony Optimization," Cluster Computing, vol. 24, no.4, pp3135-3145, 2021

[2] Nimra Malik, Muhammad Sardaraz, Muhammad Tahir, Babar Shah, Gohar Ali, and Fernando Moreira, "Energy-Efficient Load Balancing Algorithm for Workflow Scheduling in Cloud Data Centers using Queuing and Thresholds," Applied Sciences, vol. 11, no.13, p5849, 2021

[3] Ali Asghar Rahmanian, Mostafa Ghobaei-Arani, and Sajjad Tofighy, "A Learning Automata-Based Ensemble Resource Usage Prediction Algorithm for Cloud Computing Environment," Future Generation Computer Systems, vol. 79, no., pp54-71, 2018

[4] Gurleen Kaur, Anju Bala, and Inderveer Chana, "An Intelligent Regressive Ensemble Approach for Predicting Resource Usage in Cloud Computing," Parallel and Distributed Computing, vol. 123, no., pp1-12, 2019

[5] Thieu Nguyen, Nhuan Tran, Binh Minh Nguyen, and Giang Nguyen, "A Resource Usage Prediction System using Functional-Link and Genetic Algorithm Neural Network for Multivariate Cloud Metrics," Proceedings of The 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA), 20–22 November, 2018, Paris, France, pp 49–56.

[6] Chunhong Liu, Chuanchang Liu, Yanlei Shang, Shiping Chen, Bo Cheng, and Junliang Chen, "An Adaptive Prediction Approach Based on Workload Pattern Discrimination in the Cloud," Journal of Network and Computer Applications, vol. 80, no., pp35-44, 2017

[7] Rafael Moreno-Vozmediano, Rubén S. Montero, Eduardo Huedo, and Ignacio M. Llorente, "Efficient Resource Provisioning for Elastic Cloud Services Based on Machine Learning Techniques," Journal of Cloud Computing, vol. 8, no.1, pp1-18, 2019

[8] KyoungSoo Park, and Vivek S. Pai, "CoMon: A Mostly-Scalable Monitoring System for Planetlab," ACM SIGOPS Operating Systems Review, vol. 40, no.1, pp65-74, 2006

[9] Binbin Song, Yao Yu, Yu Zhou, Ziqiang Wang, and Sidan Du, "Host Load Prediction with Long Short-Term Memory in Cloud Computing," The Journal of Supercomputing , vol. 74, no., pp6554-6568, 2018

[10] Bartlomiej Sniezynski, Piotr Nawrocki, Michal Wilk, Marcin Jarzab, and Krzysztof Zielinski, "VM Reservation Plan Adaptation using Machine Learning in Cloud Computing," Journal of Grid Computing, vol. 17, no., pp797-812, 2019

[11] Jitendra Kumar, and Ashutosh Kumar Singh, "Workload Prediction in Cloud using Artificial Neural Network and Adaptive Differential Evolution," Future Generation Computer Systems, vol. 81, no., pp41-52, 2018

[12] Jitendra Kumar, Rimsha Goomer, and Ashutosh Kumar Singh, "Long Short Term Memory Recurrent Neural Network (LSTM-RNN) based Workload Forecasting Model for Cloud Datacenters," Procedia Computer Science, vol. 125, no., pp676-682, 2018

[13] Martin F. Arlitt, and Carey L. Williamson, "Web Server Workload Characterization: The Search for Invariants," ACM SIGMETRICS Performance Evaluation Review, vol. 24, no.1, pp126-137, 1996

[14] Nhuan Tran, Thang Nguyen, Binh Minh Nguyen, and Giang Nguyen, "A Multivariate Fuzzy Time Series Resource Forecast Model for Clouds using LSTM and Data Correlation Analysis," Procedia Computer Science, vol. 126, no., pp6636-645, 2018

[15] G. Prasad Babu, and A. K. Tiwari, "Energy Efficient Scheduling Algorithm for Cloud Computing Systems based on Prediction Model," International Journal of Advanced Networking and Applications, vol. 10, no.5, pp4013-4018, 2019

[16] Ramakrishnan Ramanathan, and B. Latha, "Towards Optimal Resource Provisioning for Hadoop-Mapreduce Jobs using Scale-Out Strategy and Its Performance Analysis in Private Cloud Environment," Cluster Computing, vol. 22, no., pp14061–14071, 2019

[17] Shaifu Gupta, Aroor Dinesh Dileep, and Timothy A. Gonsalves, "Online Sparse Blstm Models for Resource Usage Prediction in Cloud Datacenters," IEEE Transactions on Network and Service Management, vol. 17, no.4, pp2335-2349, 2020

[18] Deepika Saxena, Ashutosh Kumar Singh, and Rajkumar Buyya, "OP-MLB: An Online VM Prediction-Based Multi-Objective Load Balancing Framework for Resource Management at Cloud Data Center," IEEE Transactions on Cloud Computing, vol. 10, no.4, pp2804-2816, 2021

[19] Soukaina Ouhame, Youssef Hadi, and Arif Ullah, "An Efficient Forecasting Approach for Resource Utilization in Cloud Data Center Using CNN-LSTM Model," Neural Computing and Applications, vol. 33, no.16, pp10043-10055, 2021

[20] Tingting Wang, Qi Fan, Hongzhi Cai, and Beier Zhang, "Application of Machine Learning for Tracing the Origin of Metastatic Lung Cancer Tissues," IAENG International Journal of Computer Science, vol. 50, no.2, pp359-367, 2023

[21] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch, "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis," Proceedings of the third ACM Symposium on Cloud Computing, October, 2012, pp 1–13.

[22] Yin-Yin Bao, Yu Liu, Jie-Sheng Wang, and Ming-Wei Wang, "GMDH-type Neural Network Based Short-term Load Forecasting Method in Power System," IAENG International Journal of Computer Science, vol. 50, no.4, pp1194-1201, 2023

[23] Hongyu Long, Yunlong He, Wei Xiang, Zhenqi Guan, Hao Tan, and Jianbo Yu, "Research on Short-term Wind Speed Prediction Based on Adaptive Hybrid Neural Network with Error Correction," IAENG International Journal of Computer Science, vol. 50, no.4, pp1290-1304, 2023

[24] Melina, Sukono, Herlina Napitupulu, Aceng Sambas, Anceu Murniati, and Valentina Adimurti Kusumaningtyas, "Artificial Neural Network-Based Machine Learning Approach to Stock Market Prediction Model on the Indonesia Stock Exchange During the COVID-19," Engineering Letters, vol. 30, no.3, pp988-1000, 2022

[25] Padmashree G, and Karunakar A K, "Ensemble of Machine Learning Classifiers for Detecting Deepfake Videos Using Deep Feature ," IAENG International Journal of Computer Science, vol. 50, no.4, pp1279-1289, 2023

[26] Sania Malik, Muhammad Tahir, Muhammad Sardaraz, and Abdullah Alourani, "A Resource Utilization Prediction Model for Cloud Data Centers using Evolutionary Algorithms and Machine Learning Techniques," Applied Sciences, vol. 12, no.4, p2160, 2022

[27] Junhong Chen, Hong Dai, Shuang Wang, and Chengrui Liu, "Improving Accuracy and Efficiency in Time Series Forecasting with an Optimized Transformer Model," Engineering Letters, vol. 32, no. 1, pp1-11, 2024