Workload prediction for cloud services by using a hybrid neural network model

MSc Research Project
MSc in Cloud Computing

Preeti Rawat
Student ID: 20233507

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Preeti Rawat
Student ID:	20233507
Programme:	MSc in Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Shivani Jaswal
Submission Due Date:	15/08/2022
Project Title:	Workload prediction for cloud services by using a hybrid neural network model
Word Count:	6081
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Preeti Rawat
Date:	18th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Workload prediction for cloud services by using a hybrid neural network model

Preeti Rawat
20233507

Abstract

Cloud Service Providers hold vast volumes of computing resources in the cloud data centers and with advancements in the field of computing, it is possible to provide resources to the users on-demand. Automatic resource allocation shields the customers from the concerns of infrastructure needs. But there are certain issues in cloud computing related to resource management. It is important to have a resource allocation strategy to avoid resource over or under-provisioning, which in turn ensures that quality of service (QoS) is maintained as per the Service Level Agreement (SLA) between the cloud service provider and customer. Otherwise, cloud service providers have to face heavy penalties for not abiding by the SLA. To avoid these concerns a workload prediction method is proposed in this paper, which uses machine learning algorithms like Support Vector Regression (SVR) and Long-Short Term Memory (LSTM). Python language is used for the implementation of the proposed model. For conducting the experiments, a time series dataset is generated, which includes pseudo-randomness. Savitzky-Golay filter is used to remove the outliers and noise from the input signal. After that, 1 scale Wavelet Transformation is used to divide the input signal into high and low-frequency components. Low-frequency components are passed to SVR for training whereas high-frequency components are passed to LSTM for predicting computational load for the next time slot, after that by using Inverse Wavelet Transformation (IWT) output from both algorithms are combined to generate the original series. The proposed model is evaluated by measuring Accuracy(R^2), RMSE, and MSE. The metrics are calculated for two models, which are the LSTM only and the Proposed model (LSTM and SVR) and it was found that the proposed model outperformed another model.

1 Introduction

1.1 Motivation and Background

With rapid evolution, cloud data centers are growing at a high pace. The data centers are capable of handling numerous storage devices which can host numerous cloud applications. This capability requires better resource management and scheduling policies so that high scalability can be achieved in an efficient manner. Resource management is very vast and its issues require an immediate address. Resource management is the process of assigning computing processes, virtual machines, nodes, and storage resources on-demand of the application deployed in the cloud environment[1]. It helps in equal sharing of the resources between Cloud Service Providers (CSP) and users. CSP has to ensure that the services provided to the customers are at par with the signed Service level contract between them[1].

Three models of cloud service are 1) Infrastructure as a Service (IaaS) 2) Software as a Service (SaaS) 3) Platform as a Service (PaaS). IaaS providers mostly use static VM provision policies so that a fixed number of resources are allocated with a bin-packing algorithm and then there are dynamic policies that are capable of handling the load variation by live VM migration. These policies are reactive and proactive and are decided based on the knowledge of resource requirements which can be provided by the outside user or by forecasting[2].

Resource management is important for PaaS and SaaS providers so that the number of resources allotted to the various distributed applications like containers, microservices, etc can be managed. These techniques help in the dynamic scaling of resources based on the forecasted loads. Resource throttling methods can handle the thurst trends, autoscaling transients, or controlling usage of preemptible VMs. If workload information is known in advance, workloads can be prioritized in the case of the high-valued consumer.

Additionally, service orchestration helps to orchestrate the workload specialized for the target domain so that resource distribution can be made based on cost-awareness and constraints of the tasks. Also, load balancing becomes easy if load distribution is known in advance. Existing management policies are intolerant to the inaccurate prediction related to the resource requirements. The demand estimation or workload prediction can be frail therefore the scope of machine learning is more on improving it[3].

Cloud provider makes the autoscaling decisions based on certain performance criteria like hardware metrics e.g., CPU or memory usage, and certain service parameters like response time, throughput, etc. Autoscaling mechanisms are of two types reactive and proactive. The reactive mechanism helps with continuous monitoring of the system and triggers the scaling option when specified conditions are met, that is when a particular performance parameter has fallen the threshold value. But there are certain issues with the reactive mechanism, one is that reaction time can be more and it leads to overloading of the system. Additionally, if the reactive approach is used it can make the system unstable due to continuous fluctuation of the resource demand[2]. However, the proactive mechanism can help in predicting the number of resources needed for the next time period based on the mathematical model working on observed workloads. Most existing cloud service providers use the reactive model but to combat its drawbacks, research is needed in predictive models which are based on queuing theory, time series analysis, control theory, etc.

The evolution of cloud computing has led to technology trends like Fog and Edge computing which increases the level of decentralization of the computation, thereby it leads to increased variability in the resources and processed workloads. Serverless computing and edge computing helps in offloading a component of application logic at a distant location from the system core, which in turn requires thinking about resource management and scheduling. In fog and edge computing sensor data is processed across various levels of the Fog Topology. There is a need to have a mutual understanding between centralized CDC and distributed Edge computing resources so as to achieve real-time processing[3]. Such resource management methods require the placement and availability of the resources to the edge devices. It also requires taking device mobility, the topology of the network, security, and privacy. Therefore, it becomes important to study resource management in a heterogenous and decentralized environment for example fog and edge computing.

1.2 Project Specification

In this paper, a workload prediction mechanism for the cloud environment is proposed, which helps in improving the elasticity of the services provided by the cloud provider and maintaining the service quality as expected by the SLA between cloud providers and customers. A hybrid model is proposed by using machine learning algorithms like Long Short-Term Memory (LSTM) and Support Vector Regression (SVR) to handle highly fluctuating input signal. It uses techniques for canceling the noise from the input signal. At last, the performance of the proposed model is compared with LSTM for the metrics like Root mean squared error and prediction accuracy.

Research Question: Can workload prediction be improved for cloud applications by using a proactive mechanism based on composite learning on a neural network model to prevent over-provisioning or under-provisioning of cloud services by measuring the accuracy metrics like Mean square error(MSE) and prediction accuracy(R2) of the model for the computational workload?

The report is structured as follows: section 2 is the related works in the same fields done by the researchers. Section 3 discusses the research methodology to apply for the implementation of the proposed approach. Section 4 is the design specification, here architecture and proposed algorithm are discussed. Section 5 is related to the implementation of the proposed model, where key phases of the algorithm are discussed. In Section 6 evaluation, here all the experiment results are discussed by measuring the metrics and plots. Finally, in section 7, the conclusion and future work related to research work are explained.

2 Related Work

This section presents an overview of all the related work done in the same technical area. There are various workload prediction mechanisms discussed for the cloud computing system. Here workload prediction techniques in the cloud systems are discussed, along with related fields of dynamic workload prediction, autoscaling of cloud resources, etc.

2.1 Features of Support Vector Regression (SVR) and Support Vector Machine (SVM) algorithm

In the cloud data center, certain issues like task scheduling and load balancing are considered important. Authors of [4] addressed these issues and demonstrated the load balancing for virtual machines. The cloud platform providers distribute the resources based on the available resources to improve the Quality of service; this workload distribution is termed as load balancing. Load balancing is of various types like storage capacity, CPU Load, network load, and memory capacity, and it helps in maximizing resource availability, reducing the cost, and facilitating scaling and failover. The experiment was conducted in a cloud simulation environment like CloudSim and the results indicate that SVM performed better than algorithms like the random forest, naïve Bayes, and Decision Tree; and for the dynamic task scheduling, the moth flame optimization technique is used. SVM significantly reduced the processing time required for classifying the input request data and improvement in load balancing. Cloud workload is dynamic in nature and workload analysis involves the study of the type of pattern that exists in the workload like seasonal, cyclic, trend, etc. The study of these patterns helps in selecting a suitable prediction model[5].

The author of [4] has explained that by converting a multiclass classification problem into a binary classification problem it can be solved better by the binary classifier. SVM is considered the robust algorithm for designing the less complex binary classifier and it can evaluate fresh feature vector's class via hyperplanes in a high dimensional space. SVM finds an optimum separation of hyperplanes with the help of training of linear and non-linear data via SVM. SVM is performed with the distributed system with lower latency. The specification of the kernels for the SVM as per the application can improve the correctness of the SVM and moth flame optimization handles the dynamic task scheduling.

In the paper [6] authors illustrated a proactive approach for dynamically provisioning the resources for the SaaS applications by using the ARIMA model. The simulation was performed on the real traces of requests of servers from the Wikimedia Foundation. Results show the accuracy of 91%, thereby providing efficient resource utilization. It is evident that cloud workloads are quite fluctuating in nature. ARIMA is not a good choice for the prediction of fluctuating workloads. In our research after the time series sequence is sub-scaled with the help of Wavelet Transformation, SVR and LSTM algorithms are applied to the suitable sub-scale[6].

In the paper [2], time series analysis is used for creating an autoscaling mechanism to support applications showing temporal patterns. The linear statistical method can be used for time series training, which is based on autoregressive models like ARMAX, ARIMA, ARMA, etc for forecasting the CPU load from the past observations. As services parameters can be non-linear in nature and these linear statistical models cannot capture important features of the service metrics. In the study [2], it was found that SVM is suitable for the input data with linear or non-linear patterns.

As the business opportunity increases, the pay-as-you-use model can bring major changes in the usage pattern and certain technical challenges arises in capacity planning. To solve these issues a model is developed in paper [7] which can predict the cloud capacity. The experiment was conducted on trace data of IBM smart cloud. To meet the demand of the business, virtual machines are provisioned and de-provisioned quite frequently, therefore an asymmetric measure is proposed which can simulate the over or under-estimation of the cloud capacity. First, the changes in cloud capacity are divided into two parts namely provisioning and de-provisioning, finally, both components are forecasted separately. For forecasting of the provisioning component, an ensemble method for the time series is used, while, prediction of the de-provisioning part is based on the number of active VMs.

As per the authors of [7], ensemble methods join the powers of various predictors. They have given some reasons for ensemble methods: (1) The ensemble method's resilience reduces the likelihood of a significant departure from the forecast (2) On average, the ensemble method's accuracy is higher than that of a single predictor. (3) Automatically changing the prediction preference based on the cloud system's functioning condition, by altering the weights of the individual predictors. Predictors like Artificial neural networks, SVM, moving average, autoregression, and Gene expression programming are used in the experiment in the paper [7]. But the issue with these methods is in the designing of the neural network topology so that it can be efficient, additionally, the training time of the algorithm is more and become trapped in local minima. To fix these issues, the authors in paper [3] explained that the SVM regression model works well with a linear and non-linear input data pattern and it finishes with a globally unique solution with appropriate training durations. SVM is used to classify discrete data, whereas SVR is employed when doing regression. SVR operates using the same principle as SVM, of finding a best fit line or hyperplane. Therefore, SVR is used in our study for the prediction of the less fluctuating components of the input time series.

2.2 Characteristics of the Long-short term memory algorithm

Cloud Computing has challenges like dynamic rescheduling of resources and power consumption, which are being addressed in the paper [8] with the help of workload prediction by using LSTM. A study was conducted in paper [8] on the web server logs and results showed that better accuracy was achieved and mean square error was reduced to 3.18×10^{-3}. The proposed methodology not only helps in resource scaling but also in promoting green computing by reducing the number of active virtual machines. A deep learning network is comprised of special neural network layers in between eg recurrent neural networks (RNN). LSTM is a special type of RNN, they handle the issue of long-term dependency of RNNs by retaining the data for a long time.

In paper [9], to improve the workload prediction LSTM encoder-decoder includes an attention mechanism. Here, two LSTM networks work as encoder and decoder, and as an output layer. The encoder maps the historical workload as per the weights decided by the attention module into a fixed-length context vector. Context vectors are translated by the decoder into a sequence. The output layer then changes the sequence into the final output. The proposed model was evaluated in a cloud and distributed cluster environment and achieves state-of-the-art performance. A scroll prediction approach is discussed which divides long sequences into small sequences to control the prediction accuracy[9].

In the works of [10] resource usage pattern is diagnosed, to identify failure by resource contention in the cloud. First, a hybrid model using LSTM and Binary-LSTM is used to know about the future resource requirement. Lastly, the hybrid model is used to identify the state as stable or failure. The hybrid model of LSTM and BLSTM performed better than state-of-the-art methods with better predictions and lesser training times. For generating CPU and memory workloads Unix's stress tool is used, which continues to generate the same pressure for a specified period of time. To minimize the issue authors of the paper [11] suggested using a job scheduler to manage proper resource allocation among the nodes. They used a data generation pipeline that works on distributed workloads thereby generating real-time data.

The workload from the grid has high variance and its accurate prediction is difficult in the cloud system. In [12] authors have developed a model based on LSTM, which will forecast the average load over successive future time periods and the actual load many steps in advance. Real-world workload traces from Google data center and traditional distributed systems are used for the study. Traditional methods work well in the grid system and don't perform well in cloud systems; however, the LSTM model is quite adaptive to traditional grid systems and cloud systems datasets.

The approach mentioned in the paper [12] uses univariate LSTM for CPU values and uses past CPU values for the future workload prediction, However, the methodology used in [10] learns from past and future resource usage values of the time-series data for anomaly detection in a CDC. As stated in [10] due to the extra parameters of the Bidirectional Long Short-Term Memory (BLSTM) model in comparison to the LSTM model, learning takes longer. Therefore, in our research, LSTM is favoured above BLSTM.

In the study [13] authors have studied the workload dependencies on large computing systems and depending on day and time built a 2-dimensional time-series workload model. A two-dimensional LSTM neural network is designed, which supports an error propagation method. Workloads from the Shanghai Supercomputer Center are used to evaluate parallel and improved LSTM neural network model and results demonstrate that better accuracy and real-time performance were achieved. As can be seen, if LSTM is improved then it yields better performance, therefore in our research LSTM is tweaked by using the SVM algorithm so to provide better accuracy with workload prediction.

In paper [14] prediction method was applied before the actual time point so that the task scheduler has ample time to plan the jobs according to the anticipated workload. Tasks are clustered as per their nature, then models are developed for each cluster as per its characteristics. There are two types of clustering namely Prototype-Based Clustering (PBC) and Density-Based Clustering (DBC).To build the model anyone out of BRR, ARIMA, and LSTM is employed. Clustering enhanced LSTM performance, therefore, the input sequence in our research is divided into high and low-frequency clusters. LSTM trained high-frequency cluster.

2.3 The Wavelet transform

Cloud Computing and big data services are being used by the large companies. There is an increase in energy consumption in the cloud data center(CDC). To reduce the idle time of the servers, author of paper [15] suggested forecasting the short-term workload requirements of CDC so that the server load is balanced in-prior. In the works [15], multi-scale wavelet transformation is used for the input time series. The wavelet transformation is divided into wavelet decomposition and wavelet construction. Wave transformation reduces the noise from non-linear time series and filters different frequencies thereby decomposing the original series into detail and trend components. After N scale wavelet decomposition, the feature subsequence length is $1/(2^N)$. Therefore, wavelet construction is needed to regain the original series from feature sub-sequences.

In paper [16], SVM is used along with wavelet transform to predict machinery conditions. Wavelet Transformation helps in reducing the irregular characteristics and reducing the complexity of the original signal. The performance of the SVM-WT model is compared with single SVM and Neural network models, and SVM-WT outperforms both of them in effectively predicting the machinery's condition. Therefore, in our research, SVR and LSTM algorithms are used on the wavelet transformed data to perform workload prediction on the computational load.

[17] in their works have used an integrated method for workload prediction which combines Savitzky-Goley filter and wavelet decomposition with non-linear network configuration to forecast the workload for the next timeslot. A time series is first Smoothened by the SG filter, after that it is decomposed into multiple sub-features. SG filter is a data smoothing method which is called least square polynomial smoothing used to remove outliers or noise [17].

Accurate Prediction of varying workloads is important for the resource provision at the CDC. In paper [18] prediction model named BG-LSTM is proposed which can handle fluctuating workloads. BG-LSTM model comprises a bi-directional LSTM and a Grid LSTM. First workload traces are smoothed by the Savitzky-Golay filter so that extreme points and noise interference are removed. Then the integrated model based on LSTM is used for the workload prediction. Usage traces from the google data center are used for evaluating the model and results revealed that it achieves better accuracy over other algorithms for high variable cloud systems[18]. Therefore, in our research, after smoothing the input time series, LSTM is used to train highly fluctuating components.

3 Methodology

There are various methodologies available for data mining. Data mining helps in extracting meaningful information from the available data, so an adequate summary can be derived which helps in decision making. As per [19], KDD model performs quite effectively when the large dataset is involved and ML is used to extract certain insights from the dataset. In our research KDD approach is deployed.

3.1 Synthetic Dataset Generation and Storage

CDC data has a lot of alterations and it contains more fluctuations compared to the grid system. Data is generated with the help of python programming and resembles the workload of a CDC. There was no survey or experiment conducted that involved people. Dataset achieved is robust enough so that various machine learning algorithms can be applied by regression. Dataset was stored in an AWS S3 bucket, and the dataset was retrieved from S3 bucket before doing the pre-processing.

3.2 Pre-Processing of the dataset

Savitzky Goley filter removes the outliers or noisy data from the input signal. It improves the precision of the data without changing the signal tendency or preserving the peak value. As per [17] time series should be stable and requires no change around the mean value and variance of the series. Since the generated time series could have outliers, therefore SG filtering is used to have a stable time series. For smoothening of the original sequence SG filter with the window of 5 gives good results therefore, it is used in our proposed model. It is shown in Figure 1.

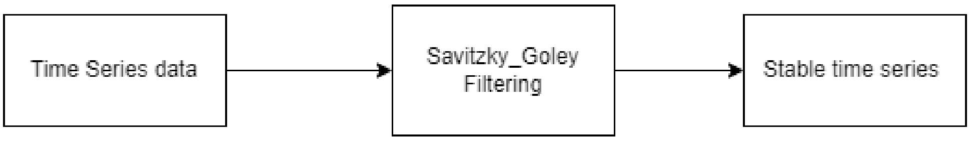


Figure 1: Smoothing using SG filter

3.3 Wavelet Transformation

Once the pre-processed signal is received, wavelet decomposition is applied so to divide the signal into high and low-frequency components. Then machine learning algorithm is applied to the respective component, so as to accurately predict the CPU workload for the next time slot.

3.4 Data Mining

It can be done after identifying the characteristics of the input signal and by applying SVR and LSTM machine learning algorithms to the low and high-frequency components respectively.

3.5 Performance Evaluation

For measuring the performance of the proposed model various performance metrics are calculated e.g. R-Square (R^2), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE).

4 Design Specification

The proposed model is implemented with the help of the python programming language. The development of the model consists of various steps like smoothing, Wavelet Decomposition, training by SVR and LSTM algorithm, and Wavelet reconstruction. This paper expands on the machine learning and Wavelet Transformation techniques employed.

4.1 The Proposed Architecture

Figure 2 describes the workload prediction strategy in a cloud system. In our research, the discussion is around the workload analyzer which forecasts the workload for a cloud data center. Sometimes there can be the scenario of uncertain workload demand in CDC, which leads to insufficient resources being allocated to process the request, which in turn impacts the quality of service provided by the cloud service provider. So, to address this issue a workload evaluator system is proposed, which can work on historical or real-time CPU workload datasets. The proposed workload prediction model uses SVM and LSTM algorithms for the input time series. Later on, a Task scheduler can be added to schedule the task based on prediction results. The resource provisioner provides the resources to the application so the overall system utilization is above the required threshold set as per SLA.

The proposed model is implemented by using the python programming language. Wavelet transformation is used to split a time series of input signals into different groups of time-frequencies. proposed model SVR and LSTM ML algorithms are used on different groups of time-frequency for the workload prediction. Lastly, the output is reconstructed and the performance of the model is calculated by measuring metrics like RMSE, MSE and R^2, these are discussed later in the coming sections.

4.2 Wavelet Decomposition

It is used to analyze the nonstationary or non-linear signals as they can reduce the non-linear component in a time series so as to improve the prediction[17]. The wavelets can extract the details of data at different scales of resolution. There are various wavelet algorithms like Daubechies wavelet, Morlet wavelet, and Mexican hat wavelet[17]. They give good resolutions for the time series that varies smoothly. But Haar wavelets outperform them as others are quite time-consuming. Also, haar uses addition instead of

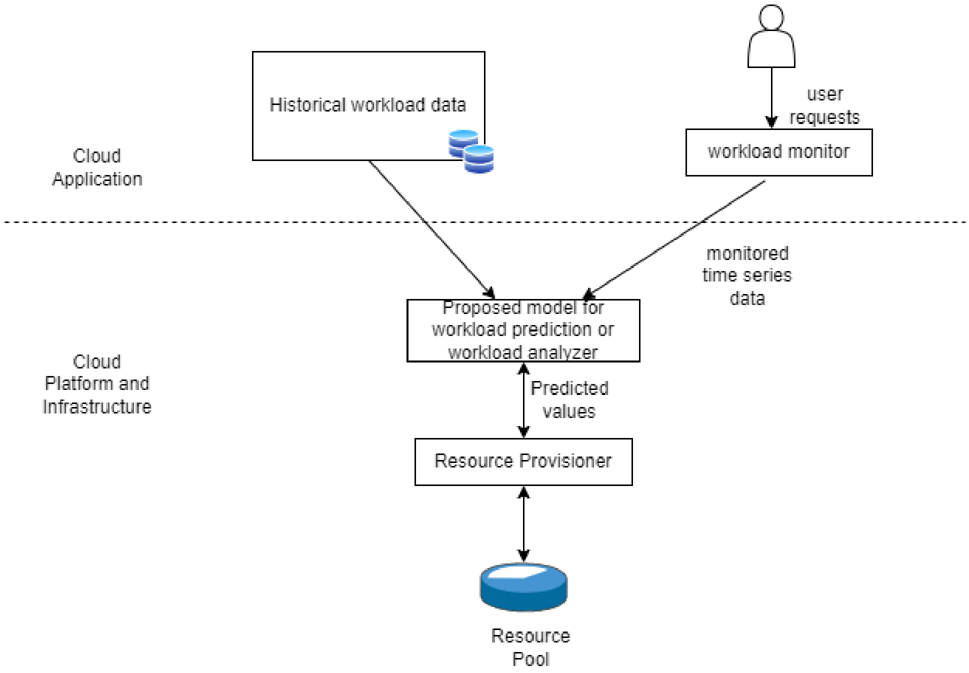


Figure 2: Workload Prediction in Cloud data centre (CDC).

multiplications and the Haar matrix has elements with zero value therefore, the computation time for the haar matrix is quite less[17].

In our research, Haar is used to extract the aspects of the time series with different resolution levels. By using One stage haar wavelet decomposition on the time series, it gets divided into two new sequences namely trend and detail. The original sequence's overall quickly changing elements are included in the trend component, which is the high-frequency component. The details components are low-frequency components that take adjacent data points of the time series into consideration, to measure the change between them.

4.3 Support Vector Regression (SVR)

In regression, a function has to be found which matches the input sample to the real numbers based on the training sample. In Figure 3, SVM components are defined. It involves two decision boundaries and one hyperplane. So, we have to consider the points which are within the decision boundaries and occupied in the margin of tolerance. The kernel is one of the parameters of the SVM and it helps in finding a hyperplane in a high dimensional space without any increase in the computational cost.

Prediction of the workload is important for autoscaling and dynamic resource scheduling. In [20] hybrid algorithm is used to improve the quality and cost of service provided by CSP. The first three-level wavelet transform is applied to the time series signal. SVR was used on two low-frequency components for the workload forecast and it was tuned by the chaotic particle swarm optimization algorithm. The High-frequency component was predicted by the GARCH model. Eventually, an ensemble method is used to recompose output from these predicted values from the multi-scale predictions, to find the workload forecast for the next time slot. The mean square error of the proposed method is 29.91% and 24.53% better than ANN and SVR respectively[20]. Therefore, in our study, SVR is

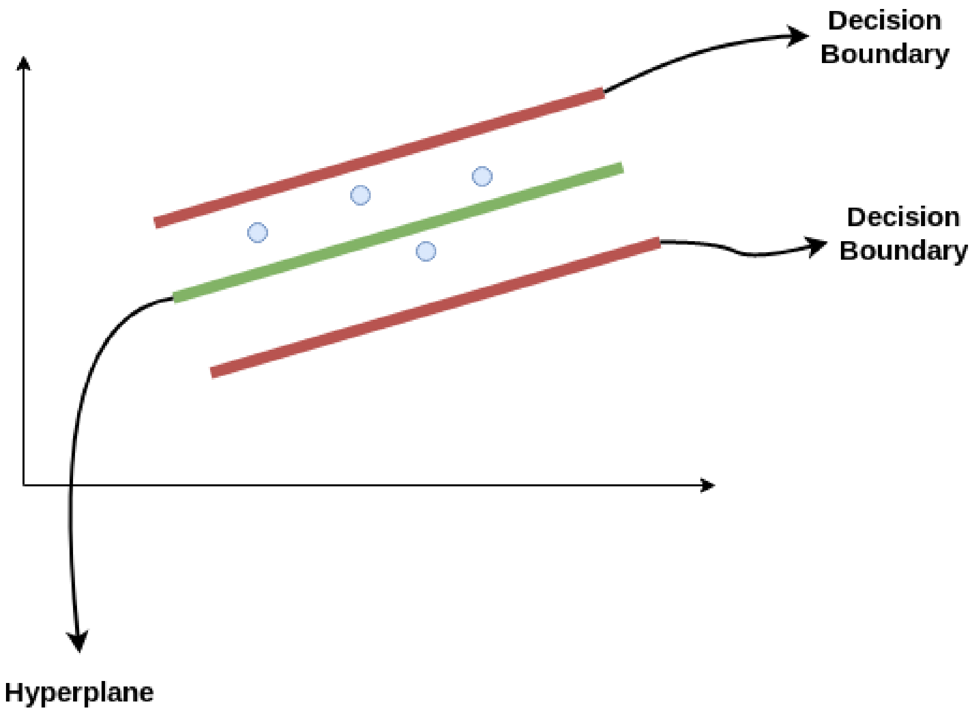


Figure 3: Components of SVR

used to train low-frequency components of the input time series.

4.4 Long-Short Term memory (LSTM)

A special form of Recurrent neural network is the LSTM. It is capable of storing the information for a longer duration of time and improves the accuracy of the model. A repeating module of LSTM is shown in Figure 4. It comprises of four neural network layers, which interact with each other. It contains two output activation functions ϕ_1, ϕ_2 and three gate activation functions $\sigma_1, \sigma_2, \sigma_3$. An important part of an LSTM cell is the cell state, which is maintained by a memory line that runs from the previous block to the current block. This line directly passes previous information to the current block but the network decides how much information should be passed. The information to be stored in the cell state is decided by two networks. Sigmoid layer σ_2 can update input value I_1. Output from tanh ϕ_2 and σ_2 layer develops a new value S_t[8]. In our research tanh is used as activation layer for the LSTM model.

4.5 Wavelet Reconstruction

Proposed model of LSTM and SVR is applied to the trend and detailed components of the time series respectively. The output signal is reconstructed from the predicted values by using Inverse Wavelet Reconstruction. In the project, it is done by using the pywt library in python.

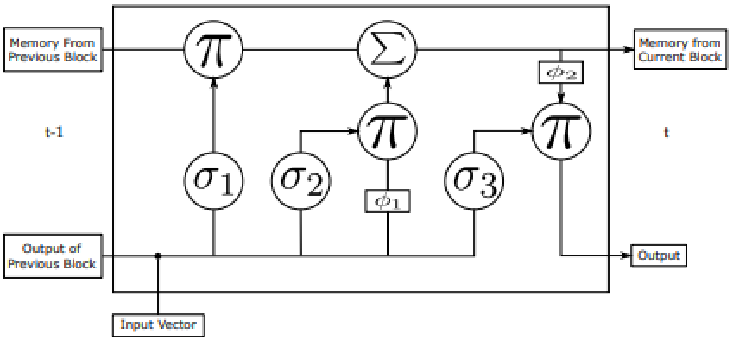


Figure 4: Repeating module of LSTM[8]

5 Implementation

Implementation is divided into four parts as shown below. Figure 5 shows the proposed algorithm workflow.

5.1 Dataset Generation and Storage

As cloud workloads are quite variable in nature. There can be more user requests in the daytime and fewer requests at night time. For checking the performance of the proposed hybrid model input data with more fluctuations is needed. So, for the study synthetic data is created with the help of python programming by including the pseudo-randomness. The wave equation can be tweaked if more fluctuations are needed in the data.

The dataset created has a size of 60,000 points. AWS S3 service is used for the storage, so that in the future if any other model is developed, the dataset can be read directly from the S3 bucket. A CSV file is created which has CPU load information and is stored in the S3 Bucket.

5.2 Smoothening and Preprocessing of input time series

The CSV file is read from the S3 bucket, then data is read from it and passed through the Savitzky-Golay filter to enhance the precision of the data without changing the tendency of the signal. It removes any outliers from the time series, which in turn removes any bias from the input series. And if outliers are not removed from the input signal, the noise gets more accentuated. This process is called smoothing.

After smoothening, the input series is divided into rows of a group of 100 values by incrementing the iteration variable with 10 values. Therefore, the input series is converted into an array of (5990, 100), which is converted into 90 values for the input and 10 values for the prediction. Next, it was divided into train and test sets. In the next stage, wavelet transformation is applied to the grouped signal.

5.3 Wavelet Transformation to decompose the input time-series

This can be one of the final stages related to preprocessing, here input series is disintegrated into low or high-frequency time series sequence. This is one of the important stages that will impact the accuracy of the proposed model. It is programmed by using the pywt (PyWavelet) library of python. Discrete wavelet transformation (DWT) is done by the dwt method of the pywt library. Haar DWT helps in signal processing by reducing the redundancy in the signal post-transformation.

After the smoothing, an array of size (5990,100) is passed for discrete wavelet transform (DWT) and there are two output series of the high frequency of size H (5990, 50) and low frequency of size L(5990, 50). Both series are divided into input series (5990, 45) and Actual Output series (5990,5) for the prediction technique.

5.4 Proposed workload prediction technique using SVR and LSTM

The described approach in the workflow is based on Support Vector Regression and Long-Short Term Memory, used for predicting the cloud workload correctly. This strategy is employed to maximize the performance of both algorithms. Since the input is comprised of high and low fluctuations components. SVR is more efficient when training low-frequency components. Frequently changing components can be handled by LSTM, as it retains the last value in memory for a long duration and provides better accuracy. When both components of the input series are modeled by both algorithms, the output is combined with the help of the Inverse Wavelet Transformation (IWT).

After passing the input signal from the Savitzky-Golay filter and performing wavelet transformation, the input signal is divided into high-frequency H (5990, 50) and low-frequency L (5990, 50) components. These components are divided into input and output sequences with sizes H(in)(5990, 45) and H(out)(5990, 5), and similarly, L(5990, 50) is divided. Both input and output sequences of H and L are divided into test and train components as shown in the Figure 5. Test dataset of low-frequency components L(1990, 45) and high-frequency components H(1990, 45) is passed to the SVR and LSTM algorithms respectively to predict CPU load values for the next five time slots. After that, the output from both algorithms is combined using IWT, and the output dataset size is (1990, 100). Out of 100 values, 90 are input and 10 are predicted values. The performance of the proposed model is evaluated by measuring metrics. Kernal used in the SVR algorithm is 'rbf' as it gives more accuracy. LSTM is implemented using the Tensorflow library in python. Also, LSTM uses an ADAM optimizer, tanh as an activation layer, and MSE as a loss function for improving the performance.

6 Evaluation

In the first experiment, LSTM is implemented individually. Then in the second experiment proposed model using SG filter, Wavelet Transformation, LSTM, and SVR algorithms is implemented successfully. After that the performance of both experiments is measured by calculating metrics like R-square, Root Mean squared Error, Mean Squared error, etc.

6.1 Experiment 1: Simple LSTM model

CPU load is read from the S3 bucket, then the input signal is smoothened with the help of SG filters, and input is divided into groups of 100 values, then it is divided into input

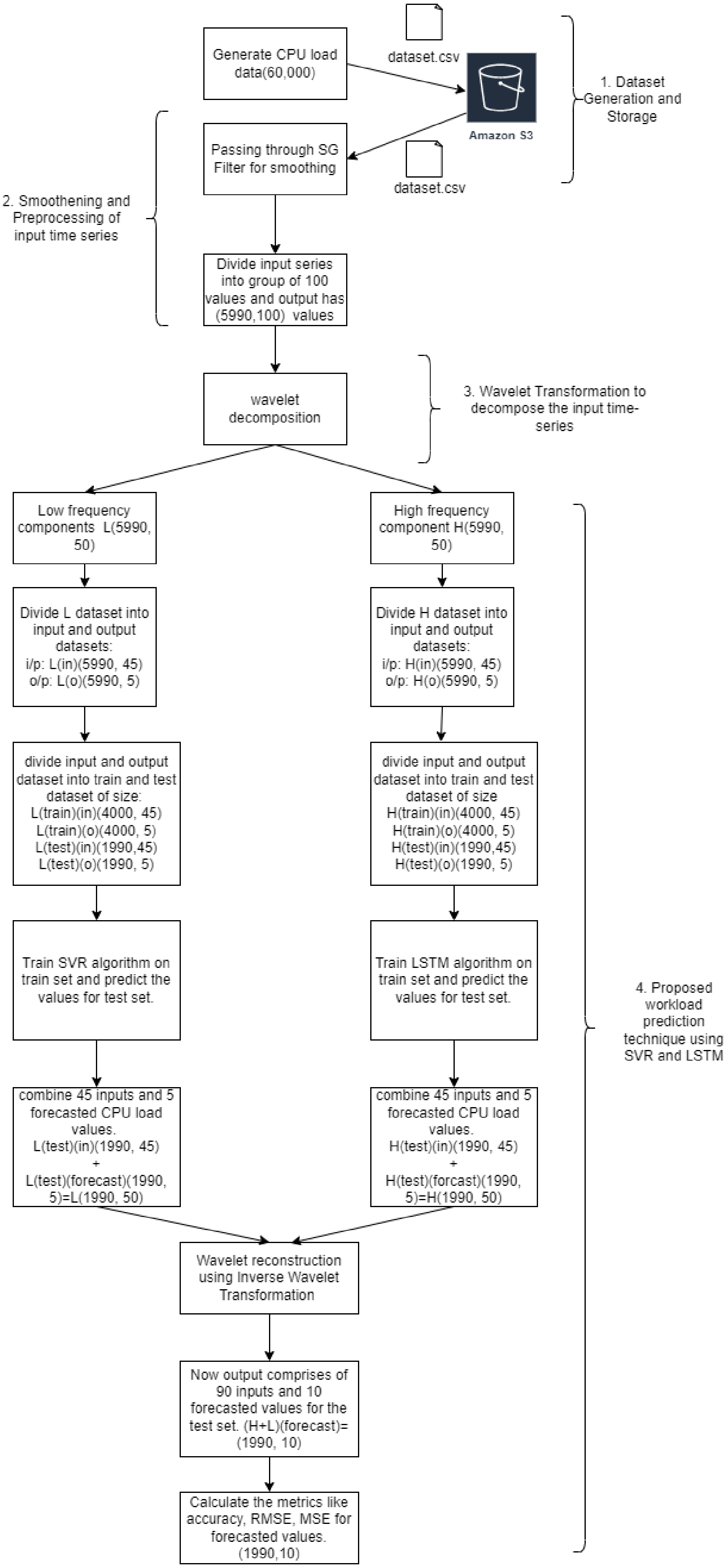

Generate CPU load data(60,000)
dataset.csv
Amazon S3
1. Dataset Generation and Storage
Passing through SG Filter for smoothing
dataset.csv
2. Smoothening and Preprocessing of input time series
Divide input series into group of 100 values and output has (5990,100) values
wavelet decomposition
3. Wavelet Transformation to decompose the input time-series
Low frequency components L(5990, 50)
High frequency component H(5990, 50)
Divide L dataset into input and output datasets: i/p: L(in)(5990, 45) o/p: L(o)(5990, 5)
Divide H dataset into input and output datasets: i/p: H(in)(5990, 45) o/p: H(o)(5990, 5)
divide input and output dataset into train and test dataset of size: L(train)(in)(4000, 45) L(train)(o)(4000, 5) L(test)(in)(1990,45) L(test)(o)(1990, 5)
divide input and output dataset into train and test dataset of size H(train)(in)(4000, 45) H(train)(o)(4000, 5) H(test)(in)(1990,45) H(test)(o)(1990, 5)
Train SVR algorithm on train set and predict the values for test set.
Train LSTM algorithm on train set and predict the values for test set.
4. Proposed workload prediction technique using SVR and LSTM
combine 45 inputs and 5 forecasted CPU load values. L(test)(in)(1990, 45) + L(test)(forecast)(1990, 5)=L(1990, 50)
combine 45 inputs and 5 forecasted CPU load values. H(test)(in)(1990, 45) + H(test)(forcast)(1990, 5)=H(1990, 50)
Wavelet reconstruction using Inverse Wavelet Transformation
Now output comprises of 90 inputs and 10 forecasted values for the test set. (H+L)(forecast)= (1990, 10)
Calculate the metrics like accuracy, RMSE, MSE for forecasted values. (1990,10)

Figure 5: Workflow of the proposed approach

signal 90 and the actual output of 10 values. After that only LSTM is applied to the (1990, 90) test set and predicts the value of the next 10 values. Figure 6 describes the comparison between actual and predicted values through a simple LSTM model. The orange color plot denotes the actual load and the blue color plot is for the predicted load. As can be seen, predicted values do not follow actual load values when the value is greater than 0.9. Figure 7 indicates the RMSE value for the LSTM model is 0.14 and MSE is 0.022. Additionally, R^2 is 0.78, this implies that the LSTM model predicts the CPU workload accurately for many time points, but does not give correct outcomes for high load values.

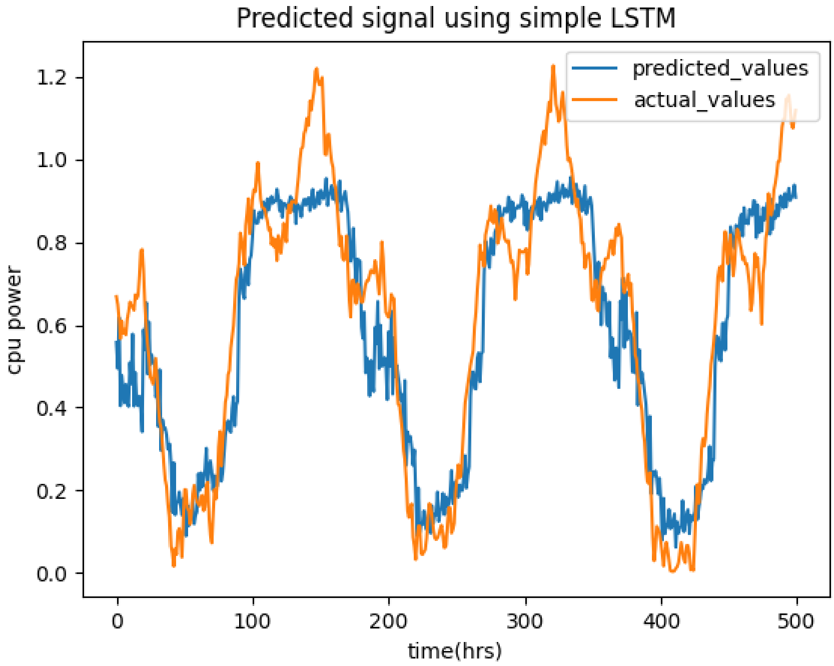
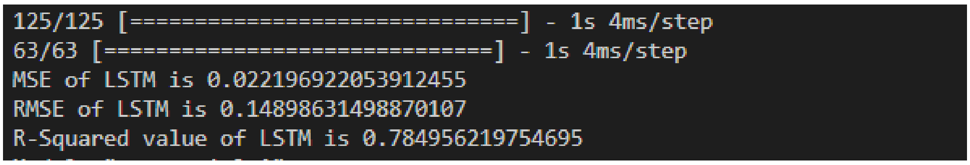


Figure 6: Predicted signal using LSTM



```
125/125 [==============================] - 1s 4ms/step
63/63 [==============================] - 1s 4ms/step
MSE of LSTM is 0.022196922053912455
RMSE of LSTM is 0.14898631498870107
R-Squared value of LSTM is 0.784956219754695
```

Figure 7: Performance metrics of LSTM

6.2 Experiment 2: LSTM +SVR hybrid model

This section provides information on how the suggested model was evaluated. In the proposed model, data generation, smoothening by Savitzy-Golay Filter, and wavelet transformation are the critical phases. Here, the input series of 90 values are divided into two sets H and L of size 45 values each. L comprises low-frequency components and SVR is used to train and predict the next 5 values. Output from SVR is combined with L then creates a 50 value set. However, H comprises high-frequency components and is passed to LSTM for training and predicting the next 5 values. Output from LSTM is combined with L, creating 50 values set. Then both new sets of 50 values are combined together to create a final output of 100 values by using inverse wavelet transformation. The final result has 90 input values and 10 predicted values.

Figure 8 shows the graphical representation of predicted and actual values for the CPU load. It shows that predicted_values follow the acutal_values for most of the time. Figure 9 shows the performance metrics of the proposed model. RMSE of the proposed model is 0.10 whereas MSE of the proposed model is 0.011. The R-squared value is 0.8840 for the proposed model, which is better compared to the simple LSTM model. Here, higher and lower CPU load values are correctly predicted by the SVR+LSTM hybrid model.

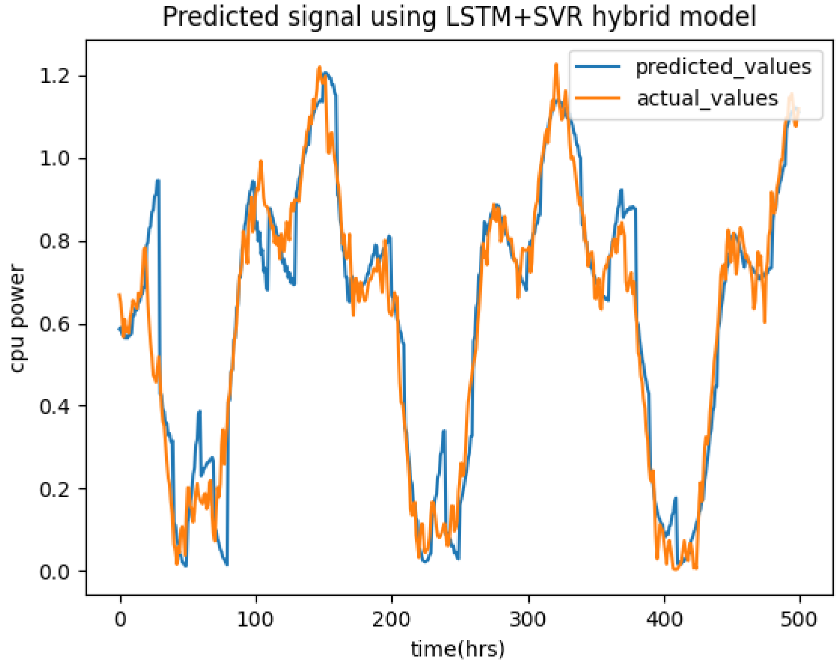
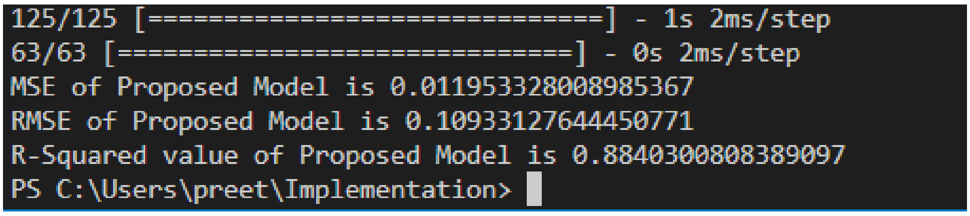


Figure 8: Predicted signal using LSTM+SVR hybrid model



```
125/125 [==============================] - 1s 2ms/step
63/63 [==============================] - 0s 2ms/step
MSE of Proposed Model is 0.011953328008985367
RMSE of Proposed Model is 0.10933127644450771
R-Squared value of Proposed Model is 0.8840300808389097
PS C:\Users\preet\Implementation>
```

Figure 9: Performance metrics of LSTM+SVR hybrid model

6.3 Discussion

Experiments conducted above have trained cloud workloads and tried to predict values for the next time slot by using simple LSTM which is a deep learning model, and a hybrid deep learning model of LSTM and SVR to work on the time series data. Proposed model has tried to bring the best out of both algorithms. Five runs were conducted for both experiments and the results are cumulated in the Table 1. As per the results, the proposed model performed well compared to simple LSTM in terms of RMSE, MSE, and R^2 values as the average accuracy of the proposed model is 0.8781 (or 87.81%), which is better than the accuracy of simple LSTM, which is 0.8015 or (80.15%). It can be observed the hybrid model was able to predict workload accurately compared to simple LSTM as simple LSTM fails for higher CPU load values.

	LSTM (Experiment 1)			SVR+LSTM (Experiment 2)		
Runs	R^2	RMSE	MSE	R^2	RMSE	MSE
1	0.76792	0.1544	0.0238	0.8803	0.1108	0.0122
2	0.7984	0.1437	0.0206	0.8719	0.1146	0.0131
3	0.7786	0.1511	0.0228	0.8792	0.1116	0.0124
4	0.816	0.1378	0.0189	0.8789	0.1118	0.0125
5	0.8467	0.1256	0.0158	0.8802	0.1111	0.0123
Average	0.801524	0.14252	0.02038	0.8781	0.11198	0.0125

Table 1: Summary of Experiment 1 and Experiment 2 runs

The experiment results are satisfactory in the workload prediction for the fluctuating and unstable input signal. Since results are derived from a different kind of test and train dataset and results achieve state-of-the-art algorithm's performance. Fluctuation of the input signals depends on the user request reaching the CDC and where the CDC is situated. It helps in improving the existing technique like LSTM since characteristics of the cloud workloads are considered, as it minimizes the error and improves the accuracy while predicting the future workload of a CDC.

7 Conclusion and Future Work

In this research, a predictive approach is assessed which is important for the autoscaling of the resources in a cloud infrastructure. Here synthetic data is generated, which involves pseudo-randomness. After that input signal is smoothened using Savitzky-Golay (SG) filters to remove any outliers. Then scale 1 wavelet decomposition is applied and the input signal is divided into low and high-frequency components. Low frequency and high components are passed through SVR and LSTM algorithms respectively. Then the output is reconstructed using Inverse Wavelet Transformation (IWT). The performance of the hybrid neural model of LSTM and SVR is compared with simple LSTM by measuring accuracy(R^2), RMSE, and MSE. Readings indicate that the proposed model outperforms the simple LSTM model and achieves an accuracy of 87.81%.

In future works, a hybrid model of LSTM and SVR can be used for autoscaling cloud resources. Also, the algorithm can be further improvised by using higher scale wavelet decomposition and other deep learning algorithms like BLSTM, ANN, GANS, etc.

References

[1] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Recent advancements in resource allocation techniques for cloud computing environment: a systematic review," *cluster computing*, vol. 20, no. 3, pp. 2489–2533, 2017.

[2] R. Moreno-Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, "Efficient resource provisioning for elastic cloud services based on machine learning techniques," *Journal of Cloud Computing*, vol. 8, no. 1, p. 5, Apr 2019.

[3] R. Buyya, S. N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, E. Gelenbe, B. Javadi, L. M. Vaquero, M. A. Netto *et al.*, "A manifesto for future generation cloud computing: Research directions for the next decade," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–38, 2018.

[4] D. Radhika and M. Duraipandian, "Load balancing in cloud computing using support vector machine and optimized dynamic task scheduling," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2021, pp. 1–6.

[5] A. Vashistha and P. Verma, "A literature review and taxonomy on workload prediction in cloud data center," in *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, 2020, pp. 415–420.

[6] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using arima model and its impact on cloud applications' qos," *IEEE Transactions on Cloud Computing*, vol. 3, no. 4, pp. 449–458, 2015.

[7] Y. Jiang, C.-s. Perng, T. Li, and R. Chang, "Self-adaptive cloud capacity planning," in *2012 IEEE Ninth International Conference on Services Computing*, 2012, pp. 73–80.

[8] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," *Procedia Computer Science*, vol. 125, pp. 676–682, 2018.

[9] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based lstm encoder-decoder network in cloud environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 274, Dec 2019.

[10] S. Gupta, N. Muthiyan, S. Kumar, A. Nigam, and D. A. Dinesh, "A supervised deep learning framework for proactive anomaly detection in cloud workloads," in *2017 14th IEEE India Council International Conference (INDICON)*, 2017, pp. 1–6.

[11] S. Kumar, N. Muthiyan, S. Gupta, A. Dileep, and A. Nigam, "Association learning based hybrid model for cloud workload prediction," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[12] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, "Host load prediction with long short-term memory in cloud computing," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6554–6568, 2018.

[13] X. Tang, "Large-scale computing systems workload prediction using parallel improved lstm neural network," *IEEE Access*, vol. 7, pp. 40 525–40 533, 2019.

[14] J. Gao, H. Wang, and H. Shen, "Machine learning based workload prediction in cloud computing," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, 2020, pp. 1–9.

[15] J. Bi, K. Zhang, and H. Yuan, "Workload and renewable energy prediction in cloud data centers with multi-scale wavelet transformation," in *2021 29th Mediterranean Conference on Control and Automation (MED)*, 2021, pp. 506–511.

[16] S. Liu, Y. Hu, C. Li, H. Lu, and H. Zhang, "Machinery condition prediction based on wavelet and support vector machine," *Journal of Intelligent Manufacturing*, vol. 28, no. 4, pp. 1045–1055, Apr 2017. [Online]. Available: https://doi.org/10.1007/s10845-015-1045-5

[17] J. Bi, H. Yuan, and M. Zhou, "Temporal prediction of multiapplication consolidated workloads in distributed clouds," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1763–1773, 2019.

[18] S. Li, J. Bi, H. Yuan, M. Zhou, and J. Zhang, "Improved lstm-based prediction method for highly variable workload and resources in clouds," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 1206–1211.

[19] J. Saltz, I. Shamshurin, and C. Connors, "Predicting data science sociotechnical execution challenges by categorizing data science projects," *Journal of the Association for Information Science and Technology*, vol. 68, no. 12, pp. 2720–2728, 2017.

[20] S. Sharifian and M. Barati, "An ensemble multiscale wavelet-garch hybrid svr algorithm for mobile cloud computing workload prediction," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 11, pp. 3285–3300, Nov 2019.