Adapted Convolutional Neural Networks and Long Short-Term Memory for Host Utilization prediction in Cloud Data Center

ARIF ullah (Aarifullahms88@gmail.com)
UTHM: Universiti Tun Hussein Onn Malaysia https://orcid.org/0000-0002-7740-2206
Irshad Ahmed Abbasi
Islamic University
Muhammad Zubair Rehman
UTHM: Universiti Tun Hussein Onn Malaysia
Tanweer Alam
Islamic Azad University
Hanane Aznaoui
Candiolo Cancer Institute

Research Article

Keywords: Prediction, Hybrid CNN- LSTM model, IaaS, Predication, Service management

Posted Date: July 9th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-597475/v1

Adapted Convolutional Neural Networks and Long Short-Term Memory for Host Utilization prediction in Cloud data center

Arif Ullah*[1] Irshad Ahmed Abbasi[2] Muhammad Zubair Rehman[3], Tanweer Alam[4], Hanane Aznaoui[5]
[1,2,3]#Soft Computing and Data Mining Centre (SMC), Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia
[4]#Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia
[5]#Cady Ayyad University, Faculty of Sciences and Techniques, LAMAI Laboratory, Marrakech, Morocco
E-mail: arifullahms88@gamil.com

Abstract

Infrastructure service model provides different kinds of virtual computing resources such as networking, storage service, and hardware as per user demands. Host load prediction is an important element in cloud computing for improvement in the resource allocation systems. Hosting initialization issues still exist in cloud computing due to this problem hardware resource allocation takes serval minutes of delay in the response process. To solve this issue prediction techniques are used for proper prediction in the cloud data center to dynamically scale the cloud in order for maintaining a high quality of services. Therefore in this paper, we propose a hybrid convolutional neural network long with short-term memory model for host prediction. In the proposed hybrid model, vector auto regression method is firstly used to input the data for analysis which filters the linear interdependencies among the multivariate data. Then the enduring data are computed and entered into the convolutional neural network layer that extracts complex features for each central processing unit and virtual machine usage components after that long short-term memory is used which is suitable for modeling temporal information of irregular trends in time series components. In all process, the main contribution is that we used scaled polynomial constant unit activation function which is most suitable for this kind of model. Due to the higher inconsistency in data center, accurate prediction is important in cloud systems. For this reason in this paper two real-world load traces were used to evaluate the performance. One is the load trace in the Google data center, while the other is in the traditional distributed system. The experiment results show that our proposed method achieves state-of-the-art performance with higher accuracy in both datasets as compared with ARIMA-LSTM, VAR-GRU, VAR-MLP, and CNN models.

Keywords— Prediction, Hybrid CNN- LSTM model, IaaS, Predication, Service management

1. Introduction

Cloud computing delivers three main services which are infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). These services are provided to the user on the basis of pay and gain rule through the internet. One of the main roles of cloud computing is to provide a huge amount of virtualized resource to the end-user [1,2]. Cloud computing main characteristics is the delivery of computation as a service in which different resources like central processing unit (CPU) , software, hardware, application are granted to user through the internet. Cloud technology has been widely functional in various fields of life and owing to its best resources on-demand delivery, low resource cost, and capricious

resource scaling. Different numbers of application have been developed on the cloud platform for improvement of these applications and different techniques are used for resource allocation and predication is one of them. But this technology is still facing several issues like resource and application balancing that can improve the performance of the system [3,4]. Resource and work-load prediction are important parameters of the cloud management systems or platforms. Prediction process improves accuracy rate and directly affects the security, quality of service, economical and management process which improve the performance of cloud computing. Normally load and application prediction are used to describe the future behavior of resources and applications on the specific aspect of the collected information base [5,6]. Figure1 shows the prediction dimension where they are used in cloud data centers.

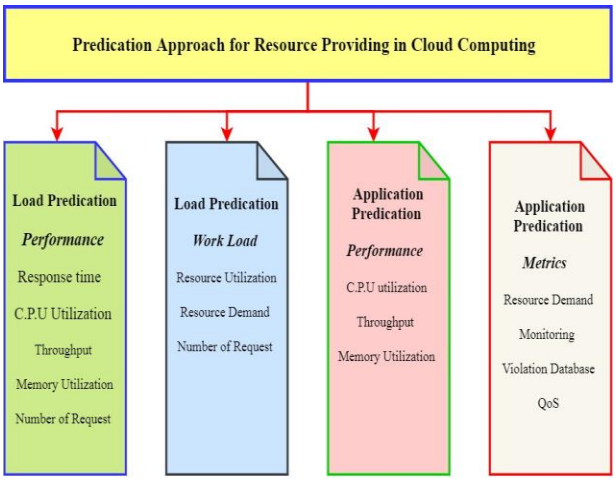

Fig.1. Predication type

Load prediction performance parameters can be measures like CPU utilization, response time, throughput, memory utilization, and network utilization. Predication approaches are future divided into application prediction and load prediction. For proactive approach, different kinds of machine learning techniques are used that uses background records of cloud applications for a particular period of time. One of the main aims of machine learning approach is to create an intelligent resource management system based on the previous data. Application prediction is one of the mandatory steps for effective resource management in cloud computing for predictions of future demands. Application prediction work in different domain like quality of service (QoS), workload prediction and service-level agreement (SLA) metrics [7,8]. To come up with future demand of resources in a fast and accurate way, prediction approach in cloud computing is important. Resource management in a cloud environment might be prophesied accurately during the application. For that reason, accurate prediction is need that can reduce the cost and manage resource usage optimally [9, 10]. IaaS provides flexible and fast information technology (IT), resources on demand, therefore, majority of the cloud providers offer scalable services that auto matically provide computer resources (such as C PU, memory, and storage). However, the scaling time to initialize a CPU and VM mainly introdu ces a delay of several minutes. To reduce scaling time delay, it is important to fix the exact amou nt of resources in advance. Consequently predict ing CPU and VM utilization is the key solution t o solve this problem [11]. The rest of the paper is organized as: The necessary background information for the resource prediction is discussed in section 2. Section 3, present the paper contribution, In Section 4: the model is proposed. In Section 5, we present the dataset information. Section 6, is based on the an alysis of the results and section 7 presents the conclusion. For the comfort of readers, we provided a list of the most frequently used acronyms in the paper in table1.

Table .1.List of Abbreviations

Acronyms	Meaning	Acronyms	Meaning
CC	Cloud computing	**MAPE**	Mean absolute percentage error
VM	Virtual machine	**HQC**	Hannan-Quinn Criterion
LB	Load balancing	**ADF**	Augmented dickey–fuller
CNN	Convolutional neural network	**MAE**	Mean Absolute Error
LSTM	Long short-term memory	**RMSE**	Mean Squared Error
DAG	Directed acyclic graph	**MSE**	Mean Squared Error

AIC	Akanke information criterion	**QoS**	Quality of service
BIC	Bayesian Information Criterion.	**FLNN**	Functional link neural network
PT	Predications technique	**CPU**	Central processing unit

2. Related work

Cloud computing is offering flexible resource allocation system on the demand of cloud customers. Establishment of the resource prediction model is difficult because the cloud user demand change over time [12]. In cloud computing, cloud server roundtrip time is important therefore in [13] author proposed neuron fuzzy network along with eight probability distribution functions for predicting the round trip time (RTT). This technique measured the time in which data travel from a source node to destination and back. The proposed technique improves the efficiency and reduces the error rate. The results the author achieved enhanced the offloading system and improve the prediction rate.

Author [14] proposed an algorithm known as swarm intelligence-based prediction approach (SIBPA) it was designed for achieving a higher prediction accuracy rate in resource allocation systems in terms of CPU, memory, and storage utilization. The proposed algorithm results are compared with well-known algorithms.

In [15], author presents multi-agent system (MAS-) for dynamic monitor prediction system for computational resource allocation system in the cloud computing. The proposed technique of reasoning agent work cooperatively with architecture system and consist of three layer sections. Multiple linear regression models were used for presenting prediction with reduced means for error system. Based on the result it author claim that it achieve good rate in error and predication role using Google platform.

Load balancing approach and balance optimization system for hardware resource utilization is important in cloud computing therefore the author in [16], proposed a model knows as long short-term memory (LSTM) encoder-decoder. The first approach is used for feature context in historical workload and the second model integrates the attention mechanism into the decoder network and carries out the prediction. All experiments are performed on Alibaba and Dinda workload traces dataset. Based on the result, the proposed technique is claim to work more accurate and small sequence monitor system.

For estimation under loaded or overloaded resource utilization in the cloud data center most of the existing estimation methods used single models technique. To address this issue the authors [17], proposed an approach by training a classifier based on statistical features system for historical resource. The proposed model is then implemented through real data set and was used for resource utilization for specific time interval. Based on the result the proposed approach achieves better results as compared with the baseline approaches.

Load balancing approach is very important for reducing resource wastage by optimizing resource utilization in the cloud data centers. Therefore the author in [18], used supervised learning technique known as support vector regression (SVRT), a technique suitable for non-linear cloud resource workload forecasting, to the future usage of multi-attribute host resources. For improvement in the training and regression section, sequential minimal optimization algorithm (SMOA) was used in the proposed technique and was implemented with different types of dataset. Based on the result the proposed technique improves 4%-16% and the error percentage was reduced by approximately 8%-60% compared with the state-of-the-art methods.

It is important to be noted that, accurate prediction of data center resource utilization needs proper planning like scheduling, energy-saving, work-load placement, and load balancing in the cloud data center. However, accurately predicting that resource utilization is a big challenge due to dynamic nature and heterogeneous infrastructures. Therefore the authors in [19], proposed a model based on deep learning adaptive window size selection methods. The sliding window size technique

captures the trend of the last resource utilization and builds an estimation for each trend period, and based on that evaluate resource utilization. The proposed estimation technique yields 16 to 54% improvement in the prediction accuracy as compared to the baseline methods.

Load balancing technique is one of the main parameters of cloud computing with the help of this technique one can improve the system life time, therefore, the author in [20], proposed osmotic hybrid artificial bee and ant colony (OHBAC) algorithm. The proposed algorithm is a combation of artificial bee and ant colony and is used to reduce number of the active virtual machines and ultimately improve the network lifetime. For resource prediction the author use simple linear regression and optimal piecewise linear regression, with the help of prediction results it accurately select best VM among all them for better utilization. The proposed algorithm improves the network stability and minimization of the system as compared to the standard algorithm.

The author in [21], proposed gradient descent (GD) and leven berg-marquardt (LM) algorithm for dynamic load prediction model of cloud computing. The proposed models are used for validation of CPU usage prediction using Google traces and its efficiency is compares with different standard models. Based on the result, the proposed models provide better results in terms of prediction.

Table .2. Summary of Related Work

Predication Technique	Platform	Metric	Pre processing	Prediction section
Hybrid neuron-fuzzy network	MATLAB/cloud-based game sessions	QoS, Communication	Yes	Server roundtrip Time
SIBPA Technique	Amazon\ Cloud data set /Cloudsim	Response time, throughput, and memory utilization predictions	Yes	Resource section
Adaptive Window Size Predictor method	Cloudsim/Alibaba/ Bit brains	C.P.U,Memory,Response Time	Yes	Resource utilization
attention-based LSTM encoder decoder network	MATLAB /Alibaba / Dinda	Prediction Accuracy	No	Workload prediction
Adaptively and automatically identify techniq ue	Cloudsim /Bit brain/ Alibaba data	CPU Utilization,	Yes	Resources Adaptiv e Prediction
SVRT technique	Bit Brain (BB), Planet Lab (PL) and Google Cluster Workload Traces	Accuracy, Reduce error Percentage	Yes	Host Resource Utilization
CNN.SVT Technique	Cloudsim/Alibaba/ Bit brains	C.P.U, Memory response time	Yes	Resource utilization
OH-BACFUP algorithm	CloudSim API 3.0.3/Cloudlets	Energy Consumption,	No	Single Resource utilization
OBD-based LM adaptation algorithm	Google cluster trace and Planet Lab workload trace/	CPU utilization,	No	CPU usage prediction

Table 2 shows the summary of related work, it consists of technique name, dataset, predication section, and platform. After the study of related work, it seems that different researchers are trying to improve the accuracy and efficiency ratio but still need improvement, therefore, this

paper carry out with hybrid technique for the improvement of predication techniques for host utilization in the cloud data center. Two prediction parameters are used to check the performance of the proposed model which is CPU and RM.

3. Paper Contribution

This section highlights the contribution made by the author's in this paper.
The paper mainly aims to optimize the cloud resource utilization by enhancing the load predication approach. Our main contributions of the paper are summarized below and Figure 2 show the working criteria of proposed model.

(1) Propose a hybrid CNN and LSTM model for multivariate resource prediction in cloud data centre.
(2) Main contribution in the proposed model is implementation of scaled polynomial constant unit activation function.
(3) The proposed model used for more host load prediction in cloud data centre.
(4) Estimation and comparison of the proposed hybrid model with different standard technique.
(5) Extensive experimental evaluation using publicly available google cluster trace and traditional distributed system data sets for different data centres in cloud environment.

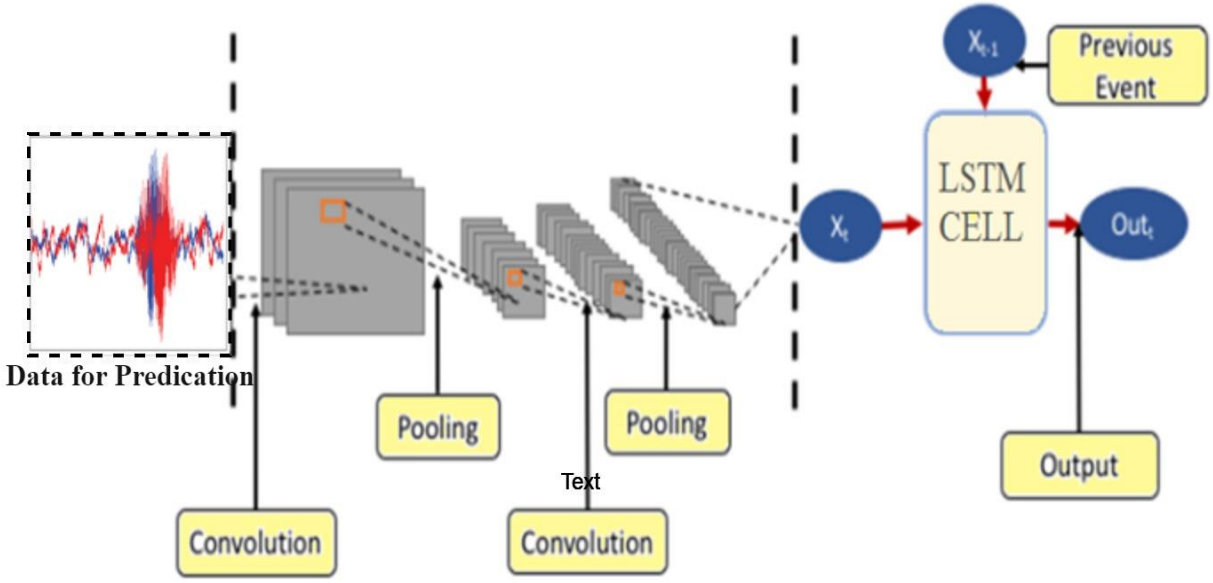

Fig.2. Structure of proposed model

4. Proposed Model

We know that CNN is good technique for removing noise and to take into account correlation between variable and multivariate and LSTM model used for temporal information and maps time serious in to detachable space to generate predictions. In our proposed model consist of CNN-LSTM which is used for predict CPU and VM utilization, VM utilization is multivariate time series that is recorded over time that including spatial information among variable and irregular patterns of temporal information. The proposed models used for resource predication the metric are CPU and RM. In the initial stage input data are analyzed by the VAR regression technique for filter the linear interdependencies among the multivariate data. After that the residual data are computed and entered in to CNN layer which extract the complex feature of each the VM usage and CPU components that LSTM temporal information of irregular in time series components and generate

the predication. In our multivariate time serious data are self-possessed of two sections these are linear and nonlinear section thus we can definite as:

$$x_t = L_t + N_t + \mathcal{E} \quad \textbf{(1)}$$

Where L_t present the linearity of data for time t used while N_t present signifies of nonlinearity section for error term $\mathcal{E}$ value used. First section for multivariate time serious x_t are analyzed by the VAR model which apprehensions the line trends. While for nonlinear or residuals of model part (N_t) used which contain spatial and temporal information [22].

$$N_t = S + T \quad \textbf{(2)}$$

In the spatial section for features and extracted with the help of CNN model and then after for inputs process hybrid CNN-LSTM model used which appropriate for modeling temporal information after that final predication generates complete. Before going to present our model we introduce some of the related of these two models after that we shift to multivariate workload prediction in cloud data center. Vector autoregressive models are design for nature tools forecasting or predication because their step are design in a such a way when current values of set of variable are partly explained it need values from the past variable then it proceed. The main role of this model to describe the joint generation mechanism of the variable involved [23,24]. The structure of variable or each variable in liner function of past logs or present logs of the other variables and itself present in the below equation.

$$y_1(t) = a_1 + w_{11}y_1(t-1) + w_{12}y_2(t-1) + w_{13}y_3(t-1) + w_{14}y_4(t-1) + e_1(t-1) \quad \textbf{(3)}$$

$$y_2(t) = a_2 + w_{21}y_1(t-1) + w_{22}y_2(t-1) + w_{23}y_3(t-1) + w_{24}y_4(t-1) + e_2(t-1) \quad \textbf{(4)}$$

$$y_3(t) = a_3 + w_{31}y_1(t-1) + w_{32}y_2(t-1) + w_{33}y_3(t-1) + w_{34}y_4(t-1) + e_3(t-1) \quad \textbf{(5)}$$

$$y_4(t) = a_4 + w_{41}y_1(t-1) + w_{42}y_2\ (t-1) + w_{43}y_3(t-1) + w_{44}y_4(t-1)\ + e_4(t-1) \quad \textbf{(6)}$$

Where the equation y1 (t), y2 (t), y3 (t) and y4 (t) are the CPU, and RM a usage present for moment t, y1 (t-1), y2 (t-1), y3 (t-1) and y4 (t-1) are the CPU,and RM movement usage t-1(here in the section the lag value is 1). a1, a2, a3 and a4 are used for constant terms, etc. h are the coefficient for the error term e1, e2, e3 and e4 are used. Before the estimate section of VAR model for the two series we specific the order p [25].Vector auto regression (VAR) model is one of the most important, flexible and essay way to analysis of multivariate time series. This model is nature postponement of the univariate autoregressive system to analysis of multivariate time series. VAR model is one the flexible for forecasts because it made the condition more future path of specified variable in the system. Vector autoregressive models used for estimation of impulse response and important preliminary step is impulse response analysis of VAR lag order [26].In this paper we resolve to use the AIC metric to estimate parameters.

$$AIC = -2ln(\hat{L}) + 2k \quad \textbf{(7)}$$

Where $ln(\hat{L})$ notation present the value of like lihood function for degree of freedom k notation is used these are parameter used in the equation. When we have model and generate AIC value small in size then they are generally better result and batter model. The residual values are calculated and arrived to the subsequent CNN-LSTM model. As the VAR model has recognized the linear trend, the lingering is anticipated to comprise the nonlinear features.

$$x_t - L_t = N_t \quad \textbf{(8)}$$

4.1 CNN-LSTM section

Convolutional neural network (CNN) is proposed from human neural system and it shows best result in wide range of application. One of the main characteristic of CNN are sparse connectivity and shared weight typical CNN is a hierarchical model they performed in computational layers like (convolutional layer and subsampling or pooling layer) and ultimately classification thought fully connected layer [27]. It is specialized type of neural network which is designed for working with different dimensional of image they may be two or three dimensional data. In the time series forecasting problem, A 1D CNN is capable of reading across sequence input and automatically learning the salient

features. A one-dimensional CNN is a CNN model having a convolutional hidden layer that operates over a 1D sequence. For time series or forecasting problem A 1D CNN is accomplished of reading across sequence number of input and inevitably learning the salient features.1D CNN is very operational for deriving topographies from a fixed length section of the overall dataset and it is not important that where the segment of data is located it work for all and work in proper section [28]. After the first layer it followed by second convolutional layer where in some cases long input sequence are equation (9) is the result of the vector y_{ij}^{1} is the output from of the first convolutions layer.

$$y_{ij}^{1} = \sigma(b_j^1 + \sum_{m=1}^{M} w_{m,j}^{1} x_{i+m-1,\ j}^{0}) \quad \textbf{(9)}$$

Where the equation section x_{ij}^{1} present the input vector section b_j^1 represent the base of j[th] feature map section w is present the kernel weight, m is the index value of filter, and σ is used for activation function like ReLU. After the convolution layer it followed by the pooling layer this layer job is to distill the output of the convolution layer to the most salient element. Main role of pooling layer is to reduce the size of the representation parameter and network computation costs. Max-pooling used for resource usage forecasting or prediction by using the maximum value of each neuron along with the cluster in the previous layer this section also effect of adjusting the over fitting section [29]. Equation 10 presents the max pooling layer.

$$p_{ij}^{1} = max y_{r\epsilon R}^{1} i \times T + r.j \quad \textbf{(10)}$$

Where T is the stride which decides that how far to movie the area of input data and R presents the pooling size that is less than the input y. After the convolutional and pooling layer followed by the LSTM layer that infers the features extracted by the convolutional section of the proposed model. Flatten layer is used between the convolutions layer and LSTM layer that is used to reduce the future maps in to single one dimensional vector [30].

4.2 Long Short Term Memory neural networks

As we know that CNN consist of different layer and LSTM is the lower layer known as CNN-LSTM which store information about power and it characteristics which are demand to extracted through with the help of CNN. LSTM provide an elucidation by antibacterial long term memory by consolidating memory units that provide information about pervious hidden state. Due to this function it become easy to find out temporal relationship on a long –term sequence and the output values from the previous CNN layer passed to the gate units. The LSTM network is well suitable for predicting power demand by addressing explosive and vanishing gradient problems. The LSTM cell comprises four interactive neural networks, each representing the forget gate, input gate, input candidate gate, and the output gate. The forget gate outputs a vector whose element values are between zero and one. LSTM network is suitable for predicting power demand by addressing explosive and evaporation incline problems [31]. The LSTM cell compare of four interactive neural networks and each representing the gate which are.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad \textbf{(11)}$$

The σ function, also denoted with the same symbol in Fig.3, is the logistic function, often called the scaled polynomial constant unit activation function. It's the activation function that enables nonlinear capabilities for the model. As we mention previously that LSTM has two property values which are hidden state $H(t)$ value of cell that change with time and $C(t)$ present the cell state which make possible to conserve memory in the long term. LSTM can add and remove information in the cell state these state are forget gate $F(t)$ for connection of input $X(t)$ for pervious hidden state $H(t-1)$ to present the cell state $C(t)$ it allow the cell to remember or forget $X(t)$ and $H(t-1)$ are used. For the input section $I(t)$ and $I(t)$ determine the feed the input value to the cell state $C(t)$.When it serves a forgetter that is multiplied to the call state it has different time step to drop values that are

not needed and keep those that necessary for predication. The output gate $o(t)$ also determines the exit based on the cell state's process is show in the below [32].

$$F(t) = \sigma\ (W_f[H(t-1), X(t) + B_f] \quad \textbf{(12)}$$

$$I(t) = \sigma\ (W_I[H(t-1), X(t) + B_i] \quad \textbf{(13)}$$

$$O(t) = \sigma\ (W_O[H(t-1), X(t) + B_o] \quad \textbf{(14)}$$

$$I(t) = \sigma\ (W_i[H(t-1), X(t) + B_i] \quad \textbf{(15)}$$

$$C(t) = F(t).\ C(t-1) + I(t).I(t) \quad \textbf{(16)}$$

$$H(t) = O(t).\tanh\big(C(t)\big) \quad \textbf{(17)}$$

The above education present the working of LSTM gate and its working criteria and figure 2 show the structure of cell and it working section. The σ function also present with the same notation in figure 3 it used fir logistic function it often called the scaled polynomial constant unit and it is the activation function that enables nonlinear capabilities for the model. Therefore in this model we change the activation function also we used the activation function. In the next step the input gate and candidate gate operate together for render the new state cell which is known as State C_t this section passed into the next step as the renewed cell state the input gate user the scaled polynomial constant unit as the activation function which is explained in the comping section and the input candidate utilizes hyperbolic tangent, each outputting i_t and C'_t .The proposed hybrid CNN-LSTM model predication algorithm work in four steps these are: Data preprocessing, fixing model, model fitting along with estimation and predication of model. As we mention before the residual value calculated by the algorithm are pass in the CNN-LSTM model. In the proposed technique four time steps are used and every sample split into pair of subsequences the CNN model can deduce every sub sequence therefore the LSTM can piece along the interpretation from the subsequences. This subsequence we split into two times as per subsequence the CNN then defined to expect two times as per subsequence with four options. Then the whole CNN model wrapped in to time distributed wrapper layer so that it applied to very subsequence the sample. After that we interpreted the result by the LSTM layer that used fifty block or neurons finally the dense layer output the predication. Figure 3 present the working of active function.

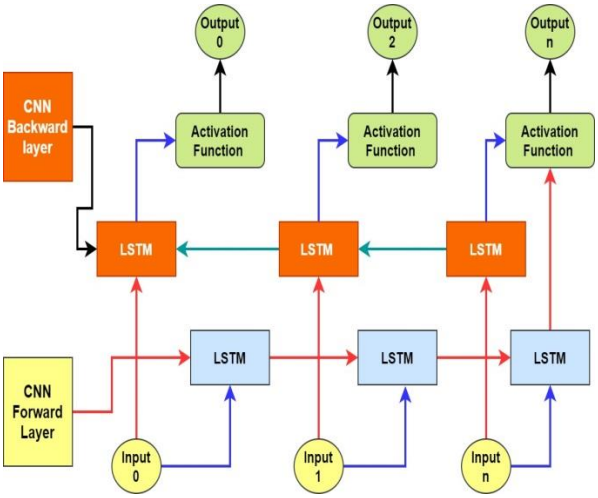

Fig.3. Activation function

The rectified Linear unit and scaled polynomial constant unit activation function is used CCN layer and LSTM block [33].

$$f(x) = x^{+} = \max(0, x) \quad \textbf{(18)}$$

Scaled polynomial constant unit activation function is activation function define by (Kise) in 2020 which is given below.

$s(x) = \alpha h\left(\frac{x}{r} + \beta\right) - \alpha h\,(\beta)$ Where $\beta \in$ (0,1) , $\alpha, r > 0\ and$

$$h(x) = \begin{Bmatrix} r(c), & x \geq c, \\ r(x), & x \in [0, c) \\ o & x < 0, \end{Bmatrix} \quad \textbf{(19)}$$

With $r\,(x) = r^3\left(x^5 - 2^{x^4} + 2\right)$ and $1 \leq c < \infty$ .we admit c goes to infinity with $r\,(c) \to \infty$ Cleary s is continuous $s\,(0) = 0\ and\ \hat{s}\,(x) = \left(\frac{x}{r} + B\right)$.Notice that for c=1 one has $h(1+) = \acute{h}(1-) = 0$ and $h(0-) = \acute{h}(0-) = 0$ which implies that $\acute{s}$ is continuous too. (This is not true for the second derivative) For c=1 the range s is the $H_{s=}[s(-\beta r), s(1-\beta)r] =$

$[-\alpha r(\beta), \alpha(1-r(\beta))], r(\mathrm{B})$ €[0,1] standard deviation algorithm is used for stochastic gradient descent for model training [34]. The network is trained for 100 epochs with batch size of 1.Where X presents the input of neuron the problem of disappearing gradient can be greatly reduce using the ReLU activation function. The network is trained for 100 epochs with batch size of 1.Where X presents the input of neuron the problem of disappearing gradient can be greatly reduce using the ReLU activation function. Table 3 present the parameters of setting of proposed model.

Table .3. Parameter setting of CNN-LSTM

Parameters	**Value**	**Parameters**	**Value**
Convolution layer filters	32	Pooling layer pool_size	1
Convolution layer kernel_size	1	Convolution layer padding Same	same
Pooling layer padding	same	Convolution layer activation function	tanh
LSTM layer activation function	SCPOS	Learning rate	0.001
Loss function	mean_MSSE _error	Pooling layer activation function	Relu
Number of hidden units in LSTM l	64	Pooling layer padding	Same
Time step	4	Batch size	64

Table 3 present the parameters setting of the proposed model and figure 4 present the Pseudocode of algorithm.

Algorithm : CNN-LSTM model training algorithm

Input: Residual: Residual values of the VAR model
N_step:The lag step between each input and output

Output: TrainPred**,** testPred: The predicted train and test data of the multivariate time series. {Phase1: Data preprocessing}

1 ze Residual data Convert into input/output with the percentage of 80%
2 Train_cl, test_cl = divide (Residual, 0.75)
3 X_train, y_train = split (train_cl, N_step)
4 X_test, y_test = split (test_cl, N_step)
5 Reshape X_train et X_test into (samples, subsequences, time steps, features)
6 **Define** model
7 **Add** TimeDistributed(Conv1D (filters = 64, kernel_size=1, activation = 'relu', input_shape = (None, N_steps, n_features)))
8 **add** TimeDistributed(MaxPooling1D (pool_size=2))
9 **add** TimeDistributed (flatten())
10 **add** LSTM (units = 50, activation = 'relu')
11 **add** Dense (n_features = 4)
{Phase3: Model fitting & estimation}
12 **Repeat**
13 **Forward propagate** model **with** X_train
14 **Forward propagate** model **with** X_train
15 **Update** model parameters
16 MSE, MAE **=** evaluate_model (X_train, y_train)
17 **If** MSE converged**:**
18 **End Repeat**
19 MSEt, MAEt **=**evaluate_model (X_test, y_test){Phase4: Model prediction}
20 TrainPred **=** predict (X_train)

Fig.4. Pseudocode of proposed model

5. Dataset information

The workload offers data arrival execution and termination of different tasks along with time stamps. In this study we analyze and predict the CPU and RM resource usage metrics. We generate and analyze the out-of-sample predictions for the succeeding 80 (50 minutes), 200 (60 minutes) and 400 (120 minutes steps ahead. Resource usage values are aggregated at 4 Seconds time interval. Google cluster trace are based on a cluster of about 12500 machine and provide information about time of different tasks arriving to the center for a 29 days period. We took 70800 samples of 7 days for training of the resources predication models and the next 30 (4 minutes) sample for time series are present as validation data for selection of appropriate parameters. Before training the network we make preprocess the data with a technique which is known as standardization by first subtracting the mean value of the training data and then dividing the standard deviation of it. Ascending approach can help with the conjunction when applied incline descent to the networks, and it noticeably improves performance of the model. During the experiment, standardization was applied to all related methods in order to obtain a fair comparison. The parameters chosen by the validating set are given in Table 3. The data set traced over 670,000 jobs and about 4 million task events across over 12500 machine darning 29 days More than ten metrics were collected by the Google trace, including CPU usage, assigned memory, page-cache memory usage, disk I/O time, disk space. As the other methods did, we only predict the CPU and RM usage values.

Table .4. Description of four load traces

Name	Description	Load traces	Mean	Standard deviation
axp7	Lightly loaded batch machine	1,123,200	0.12	0.14
axp0	Heavily loaded, highly variable interactive machine	1,296,000	1	0.54
sahara	Moderately loaded, big memory compute server	345,600	0.22	0.33
themis	Moderately loaded desktop machine	345,600	0.49	0.5

6. Experimental Results and Analysis

This section consist of different result where we present predication effectiveness of our proposed model with four exciting model like, ARIMA-LSTM, VAR-GRU, VAR-MLP, CNN and compare their predictive results.

6.1 Mean load Prediction

To make the result equivalent with other models a metric used know as exponentially segmented pattern which was used for characterize the host load fluctuation over consecutive time intervals whose lengths increase exponentially [35].The mean segment squared error (MSSE) defines as below which was applied to quantify the performance of mean load prediction.

$$MSSE(s) = \frac{1}{s}\sum_{i=1}^{n} s_i (l_i - Li)^2 \qquad (20)$$

Where s_i =b, s_i =b, 2^{i-2}, $s = \sum_i^n = 1$,b these are baseline segment , l_i is the predicated means value Li is the true value and n is the total value of segments in the predication interval.

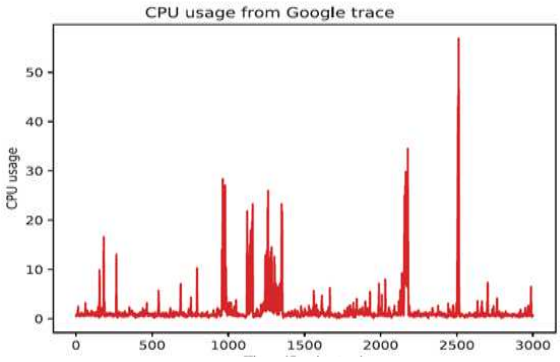

Fig.5. Mean load prediction of CPU

In the simulation first we predicted load over a single load interval and then converted it to load pattern. The proposed model result are compare with the state-of-the-art method in form of single load interval in figure 5 which predicting the mean load among consecutive future time intervals which are mention in figure 6.

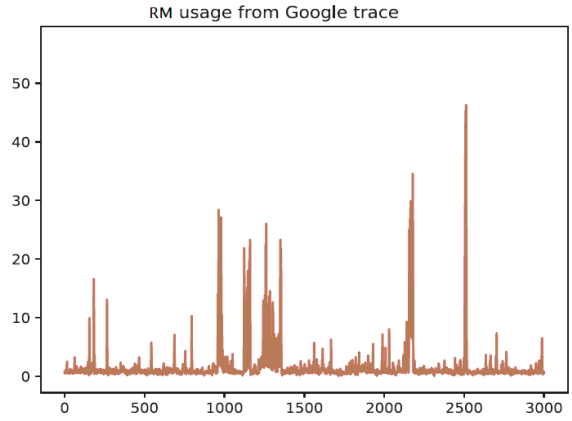

Fig.6. Mean load prediction of CPU

The details comparisons of our comparison model with other model are present in figure 6 where the mean load predications of single future interval are mention. Based on those result our model provide better result among the five single intervals due to the nonlinear generalization ability. The MSSE length is not smooth due to the high variance and noise of the host load the result of that section are mention in figure 7.

6.2 Prediction Result

The accurate predication of the C.P.U and RM utilization in cloud data center is vital to improvement of resource utilization. For this process mean-squared error (MSE) was used to evaluate the accuracy of the prediction results which is defined as below:

$$MSE = \frac{I}{n}\sum_{1}^{n}(y-y)^2 \qquad \textbf{(21)}$$

Where N is the prediction length, y is the predicted value, and $y)^2$ is the real load value. After the simulation two type of result are taking which are specific and over all result based on the result our proposed model provide an accurate prediction with history values due to the simple, regularly changes which present in figure 7.

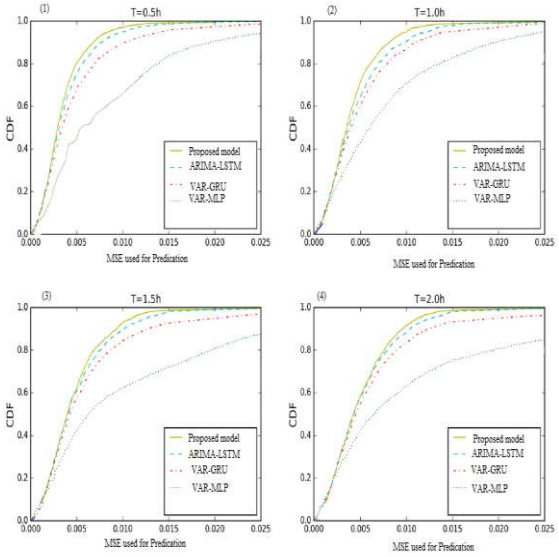

Fig.7. Result of prediction of different model

From figure 7 (1to 4) present the cumulative distribution function of (CDF) of MSE of different models the interval of Google load trace is 5 min and up to 2h is the steps ahead where a T = 0.5h. b T = 1.0 h. c T = 1.5h. d T = 2.0 h based on those result from figure 7(1to 4) our proposed model predication results are better as compare to other models.

6.3 Distribution of Load traces on System

For load trace predication we used HPC system the time services chosen here are form four most interesting host load which are axp0, axp7, sahara and themis, these are collected from load trace on Unix system collected by [36,37].These four load trace are present as diversity both in capture periods and in machine types as illustrated in Table 4 with other parameters. The load trace was scaled to a range of [0,1,0, 9] and for normalization the above formula was applied to each load trace.

$$Xi = LWB + \frac{xi - x\min}{x\max - x\min}(UPB - LWB) \qquad \textbf{(22)}$$

Where $x\max$ and $x\min$ present the maximum and minimum value of each load trace respectively LWB present the lower bound and UPB present the upper bound. The result of one of the trace is present in figure 8.

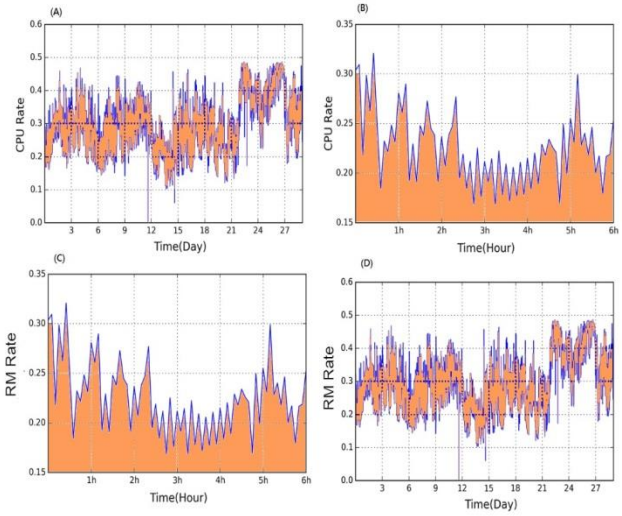

Fig.8. Normalized of load trace system

From figure 8 normalized axp7 load trace where two type of load trace are mention which are axp7 load trace and whole load trace which are mention in figure 8(a&d) and figure 8 (b&c) part of the trace which shows more details a Length of 1,200,000, b length of 200 where *x* max and *x*min are the maximum and minimum value of each load trace, respectively. LWB is the lower bound, and UPB is the upper bound. Host load predication is important element for improvement in resource allocation system in cloud computing. Due to the higher variance in data center accurate predication is important in cloud system. For that reason in this paper two real-world load traces were used to evaluate the performance. One is the load trace in the Google data center, and the other is that in a traditional distributed system. The experiment results show that our proposed method achieves state-of-the-art performance with higher accuracy in both datasets.

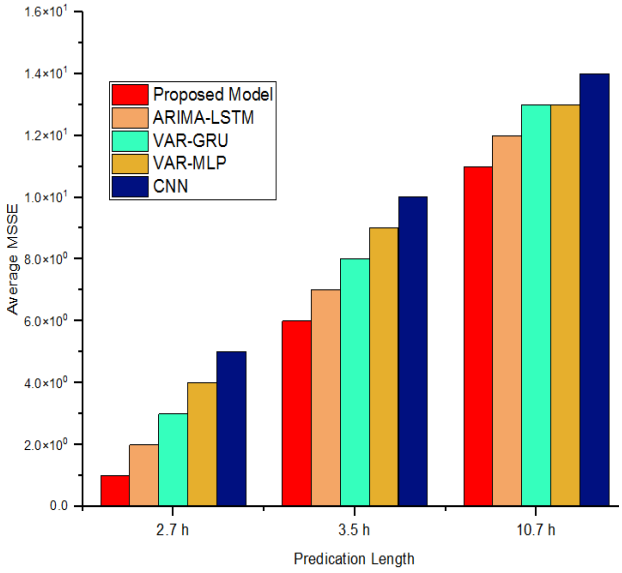

Fig.9. Predication result

We only predict the actual load value and compared with our proposed model with the four models and the original hyper- parameters of Google cluster dataset was applied to the different models to compare the generation ability of them. In this study each load trace was split 80% of its length into a training set and the rest was the testing set. The prediction results are mention in to figure 9 based on those result the proposed model indicate powerful generation base on time and execution.

6.4 Prediction Results

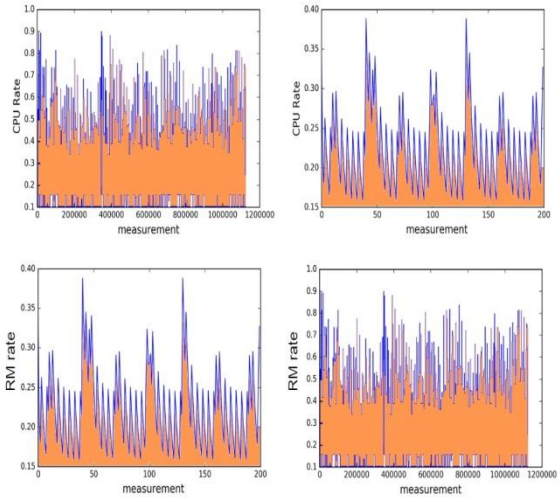

Fig.10. Predication result

Table .5. Average MSE of mean load prediction

Methods	0.5 h	1.0 h	2.0 h
ARIMA-LSTM	0.0066	0.0065	0.0050
VAR-GRU	0.0077	0.0065	0.0052
VAR-MLP	0.0076	0.0069	0.0056
CNN	0.0080	0.0072	0.0060
Proposed Model	**0.0088**	**0.0078**	**0.0064**

From figure 10 the two load prediction are mention in which actual load prediction results are mention in (Left) The load traces of axp7 on the Unix systems mention in (Right).The noisy load trace in Google cluster data drastic fluctuates. Based on those result our proposed model provide barely satisfactory performance. Table 5 present the result of MSE load predication based on the result according to the table 5 our proposed model improve the accuracy is better as compare to other models the accuracy result are 88.78% in the 0.5 h-ahead mean load prediction, by 78.06% in 0.5 h and by 64.71% in 1 h, respectively.

7. Conclusion

An important feature of cloud computing are the ability to determine allocation of resource and application based on actual usage. However for resource allocation operation required start-up time. For that reason it need plan in advance the amount of resource needed for future. For that reason in this paper we proposed new approach for host predication in term of CPU and RM utilization predication. The proposed hybrid CNN-LSTM model for multivariate workload prediction in an attempt to extract complex features of the CPU and VM usage components, then model temporal information of irregular trends in the time series components for that purpose we used new activated function. We also evaluated our proposed model with two type of dataset based on the experimental result our proposed model achieved satisfactory performance in both of the datasets. Our future work is to assimilate the proposed method in the scheduling algorithm, which will improve the resource utilization and lower the cost of the data center.

Compliance with ethical standards Conflict of interest

All authors declare that they have no conflict of interest

Funding: This paper selected for close section and has no funding.

Ethical approval: The paper not submitted any journal it the work of authors.

Reference

Sunyaev, A. (2020). Cloud computing. In Internet computing (pp. 195-236). Springer, Cham.

Tabrizchi, H., & Rafsanjani, M. K. (2020). A survey on security challenges in cloud computing: issues, threats, and solutions. The journal of supercomputing, 76(12), 9493-9532.

Jyoti, A., Shrimali, M., Tiwari, S., & Singh, H. P. (2020). Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey. Journal of Ambient Intelligence and Humanized Computing, 1-30.

Ullah, A. (2019). Artificial bee colony algorithm used for load balancing in cloud computing. IAES International Journal of Artificial Intelligence, 8(2), 156.

Iqbal, N., Jamil, F., Ahmad, S., & Kim, D. (2021). A novel blockchain-based integrity and reliable veterinary clinic information management system using predictive analytics for provisioning of quality health services. IEEE Access, 9, 8069-8098.

Ilager, S., Muralidhar, R., & Buyya, R. (2020, October). Artificial Intelligence (AI)-Centric

Management of Resources in Modern Distributed Computing Systems. In 2020 IEEE Cloud Summit (pp. 1-10). IEEE.

Hsieh, S. Y., Liu, C. S., Buyya, R., & Zomaya, A. Y. (2020). Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers. Journal of Parallel and Distributed Computing, 139, 99-109.

Biswas, N. K., Banerjee, S., Biswas, U., & Ghosh, U. (2021). An approach towards development of new linear regression prediction model for reduced energy consumption and SLA violation in the domain of green cloud computing. Sustainable Energy Technologies and Assessments, 45, 101087.

Thakkar, H. K., Dehury, C. K., & Sahoo, P. K. (2020). MUVINE: Multi-stage virtual network embedding in cloud data centers using reinforcement learning-based predictions. IEEE Journal on Selected Areas in Communications, 38(6), 1058-1074.\

Saxena, D., Singh, A. K., & Buyya, R. (2021). OP-MLB: An Online VM Prediction based Multi-objective Load Balancing Framework for Resource Management at Cloud Datacenter. IEEE Transactions on Cloud Computing.

Raza, B., Aslam, A., Sher, A., Malik, A. K., & Faheem, M. (2020). Autonomic performance prediction framework for data warehouse queries using lazy learning approach. Applied Soft Computing, 91, 106216.

Bendre, N., Ebadi, N., Prevost, J. J., & Najafirad, P. (2020). Human action performance using deep neuro-fuzzy recurrent attention model. IEEE Access, 8, 57749-57761.

Moghaddam, M. J., Esmaeilzadeh, A., Ghavipour, M., & Zadeh, A. K. (2020). Minimizing virtual machine migration probability in cloud computing environments. Cluster Computing, 1-10.

Kholidy, H. A. (2020). An intelligent swarm based prediction approach for predicting cloud computing user resource needs. Computer Communications, 151, 133-144.

Ryzko, D. (2020). Modern Big Data Architectures: A Multi-agent Systems Perspective. John Wiley & Sons.

He, Z., Chen, P., Li, X., Wang, Y., Yu, G., Chen, C., ... & Zheng, Z. (2020). A Spatiotemporal Deep Learning Approach for Unsupervised Anomaly Detection in Cloud Systems. IEEE Transactions on Neural Networks and Learning Systems.

Čeponis, D. (2021). Research of machine and deep learning methods application for host-level intrusion detection and classification (Doctoral dissertation, Vilniaus Gedimino technikos universitetas).

Kumar, J., Singh, A. K., & Buyya, R. (2020). Ensemble learning based predictive framework for virtual machine resource request prediction. Neurocomputing, 397, 20-30.

Iqbal, W., Berral, J. L., & Carrera, D. (2020). Adaptive sliding windows for improved estimation of data center resource utilization. Future Generation Computer Systems, 104, 212-224. Kumar, J., & Singh, A. K. (2020). Decomposition based cloud resource demand prediction using extreme learning machines. Journal of Network and Systems Management, 28(4), 1775-1793.

Gupta, S., Dileep, A. D., & Gonsalves, T. A. (2020). Online sparse blstm models for resource usage prediction in cloud datacentres. IEEE Transactions on Network and Service Management, 17(4), 2335-2349.

Wang, M., Xiong, S., Chen, M., & He, P. (2021). A waveform decomposition technique based on wavelet function and differential cuckoo search algorithm. Soft Computing, 25(8), 5909-5923.

Ouhame, S., Hadi, Y., & Ullah, A. (2021). An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model. Neural Computing and Applications, 1-13.

Li, X., Yi, X., Liu, Z., Liu, H., Chen, T., Niu, G., ... & Ying, G. (2021). Application of novel hybrid deep leaning model for cleaner production in a paper industrial wastewater treatment system. Journal of Cleaner Production, 294, 126343.

Yan, R., Liao, J., Yang, J., Sun, W., Nong, M., & Li, F. (2021). Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering. Expert Systems with Applications, 169, 114513.

Barrett, A. (2021). Forecasting the Prices of Cryptocurrencies using a Novel Parameter Optimization of VARIMA Models.

Thelen, A. S., Leifsson, L. T., & Beran, P. S. (2020). Multifidelity flutter prediction using local corrections to the generalized AIC. Aerospace Science and Technology, 106, 106032.

Wang, Y., Zhang, Y., Wu, Z., Li, H., & Christofides, P. D. (2020). Operational trend prediction and classification for chemical processes: A novel convolutional neural network method based on symbolic hierarchical clustering. Chemical Engineering Science, 225, 115796.

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, 9(2), 85-112.

Zhang, D., Chen, Y., Guo, F., Karimi, H. R., Dong, H., & Xuan, Q. (2020). A New Interpretable Learning Method for Fault Diagnosis of Rolling Bearings. IEEE Transactions on Instrumentation and Measurement.

Xu, Z., Li, C., & Yang, Y. (2021). Fault diagnosis of rolling bearings using an improved multi-scale convolutional neural network with feature attention mechanism. ISA transactions, 110, 379-393.

Kiseľák, J., Lu, Y., Švihra, J., Szépe, P., & Stehlík, M. (2020). "SPOCU": scaled polynomial constant unit activation function. Neural Computing and Applications, 1-17.

Mitus, A. C., Saphiannikova, M., Radosz, W., Toshchevikov, V., & Pawlik, G. (2021). Modeling of Nonlinear Optical Phenomena in Host-Guest Systems Using Bond Fluctuation Monte Carlo Model: A Review. Materials, 14(6), 1454.

Bengioy, Y., Simardy, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw, 5, 157-166.

Dinda, P. A., & O'Hallaron, D. R. (2000, May). Realistic CPU workloads through host load trace playback. In International Workshop on Languages, Compilers, and Run-Time Systems for Scalable Computers (pp. 246-259). Springer, Berlin, Heidelberg.

Load traces on unix systems. http://www.cs.cmu.edu/~pdinda/LoadTraces/ (1997)

Wu, Y., Yuan, Y., Yang, G., & Zheng, W. (2007, September). Load prediction using hybrid model for computational grid. In 2007 8th IEEE/ACM International Conference on Grid Computing (pp. 235-242). IEEE.

Yang, Q., Peng, C., Zhao, H., Yu, Y., Zhou, Y., Wang, Z., & Du, S. (2014). A new method based on PSR and EA-GMDH for host load prediction in cloud computing system. The Journal of Supercomputing, 68(3), 1402-1417.