PCU-LSTM: Predicting Cloud CPU Utilization using Deep Learning

Girish L[1], Raviprakash M L[2]

[1] Department of CSE, Shridevi Institute of Engineering and Technology , Tumakuru, Karnataka, India.
[2] Department of CSE, Kalpataru Institute of Technology, Tiptur , Karnataka, India.
[1]girishltumkur@gmail.com

Abstract –

As businesses attempt to boost flexibility and cut costs, cloud computing is becoming more popular. Despite the fact that the big cloud service providers use a pay-as-you-go pricing model and allow clients to scale up and down easily, there is still space for improvement. Workload, measured in terms of CPU utilization, fluctuates frequently, resulting in excessive costs and environmental damage for businesses. The goal of this paper is to use a long short-term memory machine learning model to forecast future CPU consumption. Companies can scale their capacity just in time and minimize excessive costs and environmental damage by estimating utilization up to 5 minutes in advance over a 30-day period. The analysis is split into two sections. The first section compares the performance of the LSTM model to a state-of-the- art model when predicting one step at a time. The second section examines the LSTM's accuracy when making predictions up to 5 minutes in advance over a 30-day period. To determine the optimal LSTM for the prediction, we compared three distinct LSTMs. To sum up, the study found that LSTM could be a beneficial model for lowering costs and eliminating unnecessary environmental effects for commercial applications hosted on the cloud.

Keywords - *Cloud Computing, Machine Learning, LSTM, Prediction.*
DOINumber:10.48047/nq.2022.20.22.NQ10194 **NeuroQuantology2022;20(22):2061-2069**

2061

1. Introduction

Data Center system is an essential rail line that influences the success of cloud computing. A scalable and efficient data centre is critical in the development and operation of reliable cloud services. The growing importance of data centre networking in recent years has put an emphasis on related issues which include connective simplification and service stability. Enterprise datacenters are often built and used for internal purposes by a single organization. This is a common occurrence among tech geeks. Colocation datacenters serve like a rental property, rendering the space and resources of a datacenter available to those who want to hire them. Managed service datacenters deliver data storage, computing, and other services to customers directly as a third party. Cloud datacenters are dispersed and are occasionally made available to consumers through a third-party managed service provider. The datacenter's engines are servers. In an edge computing architecture, the processing and memory required to manage applications on servers can be physical, virtualized, distributed among containers, or distributed among remote nodes. Processors that are most suited for the task must be used in datacenters. Both for their own purposes and for the needs of their clients, data centres store massive amounts of sensitive data. The lower the cost of storage media, the more storage space is available for backing up data locally, remotely, or both. Data access times are getting faster because too advancements in non-volatile storage media.

In recent years, data centres have seen substantial changes. Data centre infrastructure has evolved from on-premises servers to virtualized infrastructure that supports workload across pools of physical infrastructure and a multi-cloud environment as enterprise IT demands continue to migrate towards on-demand services. In the past, businesses had the idea of creating their own data centres, employing a hosting vendor, or collaborating with a managed service provider. The latter choice improved the ownership and economics of maintaining a data centre, but it did not affect the considerable lead periods for deploying and administering the equipment. Enterprises can now establish an online data centre in the cloud with just a few mouse clicks thanks to the rise of Infrastructure as a Service (Iaas) from cloud providers like Amazon Web Services and Microsoft Azure. For the first time in a single year, businesses spent more on cloud - based services than on traditional data centre systems. The local on-premises data centre, on the other hand, isn't going away anytime soon. According to a research done by the Uptime Institute in 2020, 58% of respondents admitted that the majority of their workloads remained in corporate data centers, citing a lack of visibility into public clouds as well as uptime accountability as reasons for their reluctance to make the switch.

Cloud computing is a type of utility computing that allows users to access a stream of computer processing resources such as physical machines, server software, applications, data processing, backup, networks, and other services [10] [16]. When needed, the user can use the cloud computing modality in perpetuity. In cloud computing, users often choose a mediator provider for internet service rather than setting up their own physical infrastructure. Users can only pay for the services they have used. Cloud computing has gained popularity due to numerous benefits such as low-cost storage, data availability at any time and location, and ease of maintenance. Three services are provided by cloud computing: Software as a service (SaaS), Platform as a service (PaaS), and Infrastructure as a Service (IaaS) are three types of cloud computing services [11].

Azure is a Microsoft cloud computing service that allows developers to build, evaluate, launch, and monitor apps and services utilizing Microsoft-managed data centres. Microsoft Azure provides data related services such as file storage and relational databases, as well as more specialized services like text searching and time series data. Azure SQL database and Azure Cosmos database are two of the most essential offerings. Azure SQL DB is a managed SQL server database hosting service. Azure Cosmos DB is the first worldwide cloud service to offer a multi-model database. Documents, NoSQL tables, graphs, and columnar data can all be stored and queried with it [15] [17]. Day by day, cloud services are growing more and more popular. Multiple users can share a server rack, with each user having their an isolated sandbox (virtual machine) for their application. A user can even execute an application on numerous virtual machines on different PCs. Allocating virtual machines ahead of time can improve the efficiency of underlying resources.

Time series analysis is the process of analyzing time series data and making predictions using statistics and modelling. Time series forecasting is a method of predicting future values using a model based on previously observed time series values. One of the most distinguishing characteristics of forecasting is that it does not precisely predict the future. The research employed Microsoft's cluster data trace, which details the monitoring and resource consumption information of a cluster computer throughout the course of a 30-day trace period. The dataset comprises data on Microsoft Azure's CPU utilization, which is a cloud service that is sampled every five minutes. Maximum CPU Utilization, Minimum CPU Utilization, and Average CPU Utilization are the three aspects of the data [18] [19]. This study focuses on predicting machine CPU utilization in datacenters. Separate time series detailing machine CPU utilization are derived from Microsoft Azure in our technique. Following that, an exploratory time series analysis is per- formed to better understand the nature of the time series, and two models with varied architecture are employed to address the forecasting problem.

To begin, the ARIMA [1] model is utilized, which is a tried and true traditional method for time series forecasting. The models are then compared and evaluated. The LSTM Model [2] clearly outperformed the other two models based on the test results. The error rate is only 1.004%, which is quite good and acceptable.

The Time Series and its most common methods are discussed in Section II. This section provides an overview of the most important themes. Related works are included in this category. Section III provides a more detailed and technical description of the Long Short Term Memory class of time series (LSTM). These models have three gates, which are defined as well as a more technical overview of the model identification procedure. Section IV goes into the specifics of the experiment. The first section outlines the dataset we used. The second section discusses the results. Section V summarizes the model's application to time series analysis.

2. Related Work

3. 2.1. Time series data

Time series data set is a collection of observations for a single subject (entity) at various time intervals. A single (scalar) observation is recorded sequentially across regular intervals of time in a univariate time series. Although a single column of numbers usually represents a univariate time series data collection, time is an implicit variable in the time series [13] [14]. There is more than one time-dependent variable in a multivariate time series. Each variable is dependent not only on its previous values but also on other variables. This dependency is used to predict future values. For a variety of reasons, time series data can be useful. On a daily, hourly, or weekly basis, weather data can be monitored. Medical devices that exhibit vital signs in real time are being monitored for changes in app functionality. Network logs are being tracked. A set of data points measuring a variable over an ordered time period is known as time series data. It is the most rapidly expanding database category, Although it is frequently used to understand and forecast data patterns in a wide range of industries, it's critical to check for time series elements while processing time series data for modelling. Trend is one of the components [20].

A trend is a data pattern that depicts a series movement to higher or lower values over time. A trend normally only lasts a short time before fading; it does not actually happen. Time series analysis reveals a general pattern, which is upward and indicates an uptrend, and a general pattern, which is downward and indicates a downtrend. The trend is linear or stagnant if no pattern can be discerned. Trends in data can be discovered by showing or deconstructing data sets. Seasonality is a time series property in which data varies in a predictable and consistent manner throughout the year. Seasonality is defined as any predictable fluctuation or pattern that repeats over a one-year period. Seasonality must be considered in timeseries forecasting, such as demand forecasting. The model that accounts for seasonal impacts in time series forecasting will be more accurate. In general, the purpose of time series is to develop more complicated modules by utilizing the temporal character of the data.

Correlation is a bivariate evaluation that assesses the degree of association and the path of the correlation between two variables. The correlation coefficient value varies between +1 and -1 in regards of the strength of the association. The link between the two variables will become lower as the correlation coefficient value approaches zero.

ARIMA is an acronym that stands for Auto Regressive Integrated Moving Average. Using the preceding values of a time series that can be forecasted using the ARIMA model. ARIMA is a quantitative analysis tool that employs time series data to better comprehend or forecast future trends. ARIMA is a model that combines multiple simpler models, such as the auto-regressive and moving average models. The process of regressing a variable on its previous values is known as auto regression. If there is a trend, the time series is considered non-stationary and seasonal. Integrated is a time series property that reduces seasonality. ARIMA models differ to the point where seasonality is eliminated. The moving average method eliminates non-determinism or random movements from a time series. A model that applies to lagged observations the dependency between an observation and a residual error from a moving average model. Final model will be written as ARIMA(p,d,q) p: The model's number of lag observations also known as the log order. d: The number of times the raw observations have been compared; often referred to as the degree of differencing. q: The size of the moving average window also called the order of moving average. ARIMA requires a long historical horizon, especially for seasonal products. ARIMA is assuming time series to be "Stationary". No automatic updating, high cost, unstable. As these are the drawbacks of ARIMA hence,we are migrated to LSTM to overcome these problems [9] [12].

4. 2.2. Related Papers

Francoise Beaufays et al. [3] introduced a two-layer deep LSTM RNN with a linear recurrent projection layer in each LSTM layer that outperformed state-of-the-art voice recognition performance. They concluded that single LSTM RNNs perform poorly in large-scale modelling applications. The five- layer LSTM RNN comes close to matching the top model's performance. The memorization difficulty appears to be alleviated by increasing the number of LSTMP RNN layers, which leads to greater generalization to held-out input. They discovered that adding extra layers or memory cells to the LSTM RNN models above 13M did not increase performance. They also proved for the first time that ASGD distributed training can be used to swiftly train LSTM RNN models.

An LSTM RNN-based workload prediction algorithm was presented by Jithendra Kumar et al [4]. They used an Intel Core I5-M520 CPU with a clock speed of 2.40GHz and 4GB of RAM in their tests. For implementation and simulations, Python and the Keras library were used. They looked at three different datasets: NASA, Calgary, and Saskatchewan HTTP traces. They've noticed that forecasting job load can help with resource scaling decisions when they're needed. Lastly, they found that a robust workload prediction model not only assists resource scaling decisions, but also promotes green computing by lowering the number of active computers.

According to Z.Asha Farhath et al [5], ARIMA is a prominent method for analyzing stationary univariate time series data. The three major processes in the building of ARIMA models are model identification, model estimating, and model verification, with model classification being the most crucial. This article describes an overview of the various time series prediction and forecasting models in relation to ARIMA. A vast number of real-world applications performed by a variety of persons were also examined, demonstrating that ARIMA is a useful tool for accurate time series prediction, forecasting, and analysis in the actual world.

The ARIMA model was chosen as the best candidate model for making projections for up to 5 years for sugarcane output in India using a 62-year time series data, according to Kumar Manoj et al [6]. ARIMA was chosen because of its ability to anticipate time series data with any trend and auto-correlations between subsequent values using time series data. The study also demonstrated that the fitted ARIMA time series sequential residuals (forecast losses) were not associated, and that the residuals were normally distributed with mean zero and constant variance. As a result, they concluded that the selected ARIMA (2,1,0) model is a good forecast of sugarcane production in India. Although ARIMA, like any other predictive model in forecasting, has drawbacks in terms of prediction accuracy, it is more commonly used for anticipating future successive values in time series.

Time series forecasting, as illustrated by Ratnadip Adhikari et al [7], is a rapidly growing topic of study with a lot of promise of future research. When analysing the accuracy of forecasting models, they looked at a few important performance markers. It has been recognized that in order to get a comprehensive knowledge of the total forecasting error, more than one measure should be utilized in practice.

The ARIMA model was employed by Bijeshdhyani et al[8] to analyze data from stock market time series. Using the ARIMA model, they anticipated a price that is close to the actual price on the chosen days. This indicates that the ARIMA model was applied to the time series data effectively.

5. System Model

ARIMA excels at short-term forecasting, but the system must be re-fitted each time to determine the best estimation mode, whereas LSTM succeeds at long-term forecasting. The LSTM is a sort of RNN that comprises of a collection of cells with properties that enable it to recall data sequences. In the cell, the data streams are captured and stored. In addition, the cells link one past module to another present module, allowing data to be transmitted from numerous past temporal instants to the present. Each cell's data can be discarded, filtered, or added towards the next cell's data.

3.1. Related Papers

***Data Collection**:* The data for training and testing the model was obtained through Microsoft Azure, a cloud service. Microsoft introduced it to develop, distribute, and manage applications and services across a worldwide network of Microsoft-managed data centres. The data is from January 1st to January 30th of the year 2017.

Data Preprocessing: The main aim of this phase is to convert the historical workload traces into the format required by the suggested model. It is crucial to remove any noisy data from the accessible dataset traces before applying to any models.

Testing and Training: In the ratio 80:20, we divided our sample into testing and training data.

Model training using LSTM: ARIMA was used as a benchmark, and an LSTM model was built on top of it. LSTM has indeed been demonstrated to be more good at predicting time series data than other machine learning models. The RNN method underpins the LSTM's operation. The LSTM model has the ability to remember and capture the best elements over a lengthy period of time.

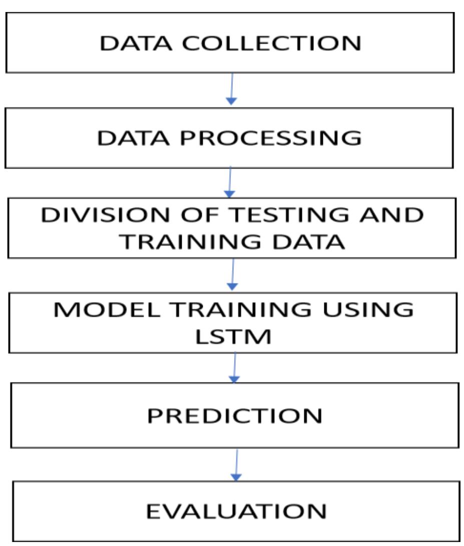

Fig. 1 Flow Diagram of the Proposed System

Prediction and Evaluation*:* The accuracy of the forecast was evaluated employing three evaluation criteria: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Average Percentage Error (MAPE) . In terms of percentage accuracy, a lower MAPE score indicates better prediction accuracy.

3.2. Long Short Term Memory

LSTM, a kind of recurrent neural network that can acquire order dependence. Because significant occurrences in a time series can have unforeseen lags, LSTM networks are quite well to classifying, evaluating, and drawing conclusions based on time series data. Because of its decreased susceptibility to spacing, LSTM surpasses RNNs, hidden Markov models, and other learning algorithms in many contexts. LSTMs are capable of promptly dealing with two types of technical issues: vanishing gradients and exploding gradients, both of which are related to the network's learning phase. The difficulty is that as an input cycles through the network's recurrent connections, its impact on the hidden neurons, and hence on the network output, diminishes or grows exponentially.

LSTM, a kind of recurrent neural network that can acquire order dependence. Because significant occurrences in a time series can have unforeseen lags, LSTM networks are quite well to classifying, evaluating, and drawing conclusions based on time series data. Because of its decreased susceptibility to spacing, LSTM surpasses RNNs, hidden Markov models, and other learning learning phase. The difficulty is that as an input cycles through algorithms in many contexts. LSTMs are capable of promptly dealing with two types of technical issues: vanishing gradients and exploding gradients, both of which are related to the network's the network's recurrent connections, its impact on the hidden neurons, and hence on the network output, diminishes or grows exponentially.

1. Forget Gate: The forget gate is only given crucial information about the lstm unit is the preceding camouflaged state and is the current input, and the addition of both is evaluated using the sigmoid activation function, which converts the output value from 0 to 1.

$(h_{t-1} + x_t)$ (1)

h_{t-1} (2)

x_t (3)

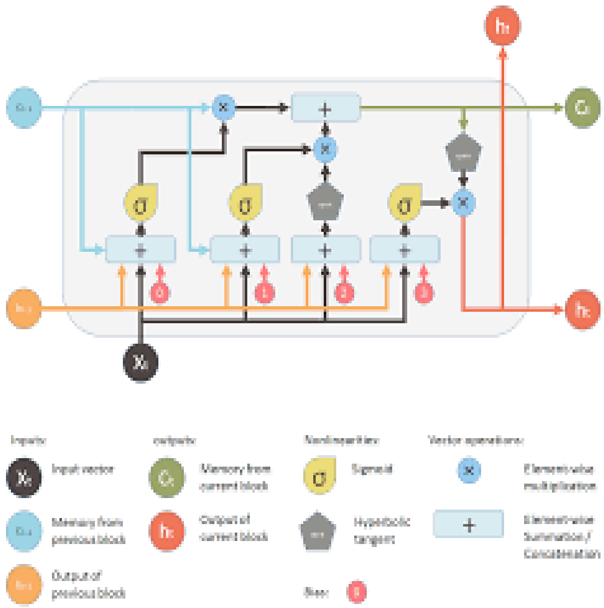

Fig. 2. Long Short Term Memory (LSTM) Model

2. Input Gate: The activation function, which is generally a Sigmoid activation function in the range of 0 to 1, evaluates and provides extra input, just as the forget gate. The input gate input is generated and is processed with the Tanh activation function, which returns a value between -1 and 1. This new input data is multiplied by the candidate's input.

$(h_{t-1} + x_t)$ (4)

(i_t) (5)

(c_t) (6)

(c_t) (7)

$(c_t) * (i_t)$ (8)

This data is also transmitted to the cell state.

3. Cell State:. The data from the forget gate is coupled with the data from the input gate, yielding the equation.

$(c_t) * (f_t)$ (9)

$(c_t) * (i_t)$ (10)

Cell state = $(c_{t-1}) * (f_t) + (c_t) * (i_t)$ (11)

Because it contains all essential data, the cell state works as a memory unit for the LSTM structure, as shown in the equation.

4. Output Gate: Under the Sigmoid Activation function, the cell state information is multiplied with the output gate processes, condensing the information in the 0 to 1 range.

$(h_{t-1} + x_t)$ (11)

This cell state data is processed by the Tanh Activation function, as well. The Hadamard product, which is simply element-wise multiplication, is used for every multiplication. Cell state data is multiplied by the output data.

(o_t) (12)

5. Hidden State output: Hidden state output is created by multiplying cell state and output information.

$h_t = ((c_{t-1}) * (f_t) + (c_t) * (i_t)] * (tanh) * (o_t))$ (13)

In several models, activation functions such as Softmax, ReLU, and leaky ReLU activation functions have been shown to assist avoid the vanishing gradient problem. The workings of both layers, LSTM and GRU, reveal that GRU uses fewer training parameters, requires less memory, and runs faster than LSTM, while LSTM is more accurate on larger datasets. LSTM is employed when dealing with lengthy sequences when accuracy is a concern.

6. Results and Discussion

This section contains information on softwares and datasets used. NumPy, a Python library for array processing, was one of the libraries we loaded. It offers a high-performance multidimensional array object and utilities for working with these arrays. Matplotlib inline supports inline plotting, which displays plots or graphs immediately below the cell where you write your plotting comments. Pandas is a library that allows us to examine large amounts of data and draw conclusions based on statistical theories. It has data analysis, data-cleaning, data explanation, and data manipulation functions. TensorFlow is an open-source numerical calculation library written in Python that makes machine learning faster and easier. Import CV2, a python bindings package for solving computer vision challenges. PIL Import Image, the application loads an image, creates a blurred version of it and saves the resulting image to disc. TQDM, which progress bar not only displays the amount of time that has passed but also the estimated time remaining for the iterable.

4.1. Dataset

The data on which we base our work was obtained from Microsoft Azure, which is a cloud service. The Dataset con tains the CPU Usage sampled every 5 minutes. It has the data of 8641 compute clusters in five-minute intervals during 30 days. The data has three attributes Max CPU Utilization, Avg CPU Utilization, and Min CPU Utilization.

4.2. Results

For the experimental evaluation, we have considered the mean absolute error loss function. As shown in Table 1, the prediction error was lower in the stacked LSTM, when compared to the unidirectional Vanilla LSTM and Bidirec- tional LSTM models, the Stacked LSTM model performs exceptionally well. The stacked LSTM model had a validation loss of 0.0291.

Table 1. Training and validation loss with different LSTM models

Type of LSTM	Training Loss	Validation Loss
Vanila LSTM	0.0577	0.0324
Stacked LSTM	0.0803	0.0291
Bi-Directional LSTM	0.0840	0.0336

For the model construction, we have used two LSTM layers for both Bi-Directional LSTM and Stacked LSTM with a batch size of 512, and a dense layer is used after the LSTM layer as an output layer. Fig 3 shows the summary of the proposed stacked LSTM model. During the literature survey, we found that majority of the researchers have used ARIMA and RNN models for the forecasting applications. When compared to RNN, LSTM networks are an ideal model for predicting workloads because they consume less computational resources and produce more accurate forecasting results than statistical models like ARIMA and RNN. Our proposed model provides better performance than the existing models.

Fig 4 and Fig 5 describes the graphical representation of Testing and Predicting values of Minimum and MaximumCPU Utilization.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
lstm_2 (LSTM)               (None, 500, 512)          1056768
_________________________________________________________________
lstm_3 (LSTM)               (None, 512)               2099200
_________________________________________________________________
dense_1 (Dense)             (None, 3)                 1539
=================================================================
Total params: 3,157,507
Trainable params: 3,157,507
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3. Model Summary of Stacked LSTM

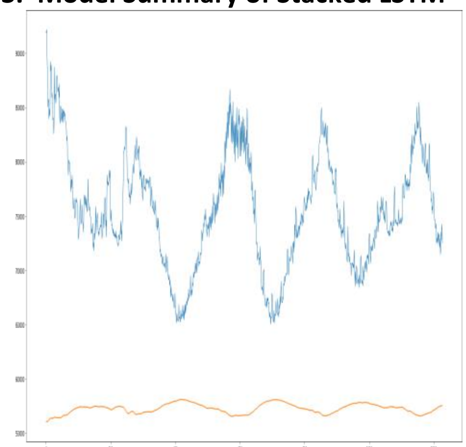

Fig. 4. Minimum CPU Utilization

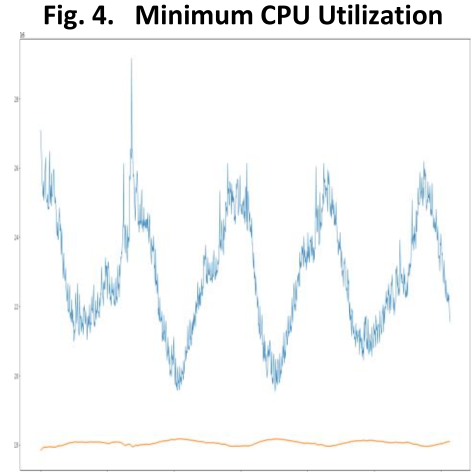

Fig. 5. Maximum CPU Utilization

2067

7. Conclusion

SARIMA model outperformed the other models when it came to forecasting long-term trends, but having poorer short- term performance. More fitting resources, a hard effort macro component optimization, and the use of actual data without downsampling would vastly enhance the precision. Although formulating periodical forecasts with the SARIMA model may seems unfeasible, it can be used to examine Long-term performance. Earlier cloud resource research study has underscored the importance of projecting long-term usage in order to calculate how consumption of resources would fluctuate. The core objective, according to the Microsoft Azure researcher, is to forecast the largely driven CPU utilisation over the next hour and scale up (or scale down) their clusters to accommodate those needs. The determination to scale up or down is based on a risk appraisal that calculates the relative costs of idle cluster resources vs the demand to swiftly increase resources to meet unexpected demand, as well as the time it takes to allot and close a deal systems.

The algorithm that is attuned to making repeated predictions over a one-hour period will be more effective for this project. The LSTM approach, which is shown in this study, can more accurately predict upcoming CPU usage for the next hour in terms of both efficiency and durability. This study will be used as a framework for new cluster resource prediction installations. The goal of this study was to use LSTM to forecast cluster CPU usage. The LSTM surpasses the ARIMA in this research. Finally, the findings show that LSTM could be a beneficial tool for lowering costs and decreasing unnecessary environmental effect for commercial applications hosted in the cloud.

References

[1] Langston Nashold, Rayan Krishnan "*Using LSTM and SARIMA Models to Forecast Cluster CPU Usage*," published:16 Jul 2020.

[2] Pranav T P, Charan S, Darshan M R, "*Devops Methods for Automation of Server Management using Ansible* ", International Journal of Advanced Scientific Innovation, vol. 1, no. 2, pp. 7-13, May 2021.

[3] Sak, Has¸im, Andrew Senior, and Franc¸oise Beaufays. "*Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition*." arXiv preprint arXiv:1402.1128 (2014).

[4] Jitendra Kumar,Rimsha Goomer,Ashutosh Kumar Singh "*Long Short Term Memory Recurrent Neural Network(LSTM-RNN)Based Workload Forecasting Model For Cloud Datacenters"*.Published:7-8 Dec 2017.

[5] Z. Asha Farhath,B.Arputhamary,Dr.L.Arockiam,"*A Servey On ARIMA Forecasting Using Time Series Model*".Published:8 Aug 2016.

[6] KUMAR Manoj,ANAND Madhu,"*An Application Of Time Series ARIMA Forecasting Model For Predicting Sugarcane Production In India*".

[7] Ratnadip Adhikari,R.K. Agarwal,"*An Introductory Study On Time Series Modeling And Forecasting*".DOI:10.13140/2.1.2771.8084.

[8] Bijeshdhyani,Manish Kumar,Poonam Verma,Abhishik Jain, "*Stock Market Forecasting Technique Using ARIMA Model*".Published:6 Mar 2020.

[9] L. Girish and S. K. N. Rao, "*Mathematical tools and methods for analysis of SDN: A comprehensive survey,"* 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016, pp. 718-724, doi: 10.1109/IC3I.2016.7918055.

[10] Anupama K C,Shivakumar B R,Nagaraja R,"*Resource Utilization Pre- diction in Cloud Computing using Hybrid Model*".Vol 12.No.4,2021.

[11] Khanum, Salma, and L. Girish. "*Meta heuristic approach for task scheduling in cloud datacenter for optimum performance*." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 4.5 (2015): 2070-2074.

[12] Girish, L., Rao, S.K.N. "*Anomaly detection in cloud environment using artificial intelligence techniques*". Computing (2021).https://doi.org/10.1007/s00607-021-00941-x

[13] Prajwal S, Siddhartha M, Charan S. (2021). *" DDos Detection and Mitigation SDN using support vector machine".* International Journal of Advanced Scientific Innovation, 1(2), 26-31. https://doi.org/10.5281/zenodo.4782280

[14] Girish L, Rao SKN (2020) *"Quantifying sensitivity and performance degradation of virtual machines using machine learning"*. J Comput Theor Nanosci. https://doi.org/10.1166/jctn.2020.9019

[15] Rashmi TV, Prasanna MK, Girish L (2015)" *Load balancing as a service in Openstack-Liberty*". Int J Sci Technol Res 4(8):70–73

[16] Raviprakash M L, C S Pillai, Ananda babu J, "*Object Recognition using Lucas-Kanade Technique and*

Support Vector Machine Based Classification in Video Surveillance Systems", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019.

[17] Ravi Prakash M L , Chethan Chandra S Basavaraddi , Ananda Babu J, "*Face Recognition Using Hybrid Neuro Fuzzy Network for Occluded Images*", International Journal of Science and Research (IJSR), 2018

[18] Girish L, & Raviprakash M L. (2022). Data analytics in SDN and NFV: Techniques and Challenges. International Journal of Advanced Scientific Innovation, 4(8). https://doi.org/10.5281/zenodo.7657569

[19] Chethan Chandra S Basavaraddi, Raviprakash M L, Dr. C S Pillai, *"Face Recognition from Feed Forward Neural Network Using Occluded Images for Automating the Surveillance Using Hybrid Neuro Fuzzy Network",* International Journal of Engineering Applied Sciences and Technology, 10.33564/ijeast.2020.v05i02.087, 2020, Vol 5 (2) pp. 508-519

[20] Raviprakash M L, C S Pillai, "*Histogram Specification and Kalman Filter with Euclidian distance Technique for identification of Human in the video Surveillance systems*", IJRAR March 2019, Volume 6, Issue 1.