Jimma University

Institute of Technology,

Faculty of Computing and Informatics

Program-Computer Networking.

Enhanced Levenberg Marquardt Algorithm for Workload Prediction of Cloud Virtual Machine.

By: Adugna Tsegaye Duresso

A Thesis Submitted to The School of Graduate Studies of Jimma University in Partial Fulfillment of The Requirement of The Degree of Master of Science in Commuter Networking.

Advisor: Girum Ketema (PhD)

Co-advisor: Admas Abtew (PhD candidate)

Ethiopia, Jimma

Dec, 2021.

Approval Sheet

Jimma University

School of Graduate Studies

As thesis research advisor, we here by certify that **Adugna Tsegaye** thesis work entitled" **Enhanced Levenberg Marquardt Algorithm for Workload Prediction of Cloud Virtual Machine"** under our guidance.

I recommended that it submitted as fulfilling the thesis requirement

Dr Girum Ketema (PhD)	____________________	____________
Advisor	*Date*	*Signature*
Mr Admas Abtew (PhD Cand)	____________________	____________
Co-advisor	*Date*	*Signature*

As member of the board of examiners of the master of science thesis open defense examination, we examined the thesis prepared by **Adugna Tsegaye** and certified the candidate. We recommended that the thesis accepted as fulfilling the thesis requirement for Degree of Master of Science in Computer Networking.

Dr Melkamu Deressa (PhD)	____________________	____________
External Examiner	*Date*	*Signature*
Sofanit Alem (MSC)	____________________	____________
Internal Examiner	*Date*	*Signature*
Ermiyas Tesfaye (MSC)	____________________	____________
Chairperson	*Date*	*Signature*

DECLARATION

I declare that the thesis work entitled **"Enhanced Levenberg Marquardt Algorithm for Work-load Prediction of Cloud Virtual Machine"** is my and under guidance of **Dr Girum Ketema and Mr Admas Abate.** All sources of material used for this thesis have dully acknowledged and submitted to Jimma university in partial fulfillment of the requirement for the award of master of Computer Networking.

I also hereby declare that this work in partial fulfillment has not submitted to any other university for any Degree or Diploma

Submitted By

Adugna Tsegaye ___________________________ _____________________

Researcher's Name: *Date* *Signature*

This thesis has submitted for examination with my approval as university supervisor

Dr Girum Ketema (PhD) ___________________________ _____________________

Advisor *Date* *Signature*

Mr Admas Abtew (PhD Cand) ___________________________ _____________________

Co-advisor *Date* *Signature*

Acknowledgment

First and foremost, I wish to express my sincere appreciation to my Advisors, Girum Ketema (PhD) and Basilos Tilahun (PhD candidate) for their continuous guidance, and encouragement throughout the course of this research.

Secondly, I also wish to thank my co-advisor, Mr Admas Abtew, for His support to complete this research.

Finally, I would like to say thanks for my family, friends, and colleagues.

Abstract

Cloud computing is the fast-growing technology which had been using widely for distributing and acquiring the required resources through internet. Due to incoming workload to cloud have nonlinear and inconsistent characteristics, predicting future workload remains a critical task. This causes, attempting to reduce numbers of running server at cloud data center also the key challenge in providing cloud services. One of the possible methods to overcome the issues is having prediction model which can increase accuracy level in dynamic workload environment. Purpose of future workload prediction is to provide reduction time window for deployment of cloud physical servers and virtual machine creation and their allocation.

Neural network supervised machine learning model used for learning and predicting future workload. Levenberg Marquardt algorithm is the combination of the steepest descent with low convergence and Gauss newton method with opposite characteristics. The major drawbacks of Levenberg Marquardt algorithm, it assumed independent of its initial point without proper initial weight initialization. Lack of method lead the model to many problems like, dead neurons, gradient disappearance, accuracy, and convergence problem. To increase the workload prediction of virtual machine accuracy level, this paper designed proper weight initialization techniques for 'LM' and implement in MATLAB R2016a. To show the performance of proposed system, the result evaluated against 'RNN' algorithm where historical cloud virtual machine log data used for training in which CPU and Memory intensive workload used for performance metrics. From obtained result, we concluded the method show better accuracy than existing method in dynamic incoming workload to cloud datacenter.

Keywords: Levenberg Marquardt algorithm, Workload Prediction, Prediction Accuracy, Machine Learning, Enhanced LM, and Prediction Model.

Table of Contents

Table of Figure

List of Tables

List of Acronym

NN	Neural Network
ANN	Artificial Neural Network
BNN	Backpropagation Neural Network.
LMBNN	Levenberg–Marquardt Back Propagation
LM	Levenberg–Marquardt
Bp	Backpropagation
ML	Machine Learning
SaaS	Software as a Service
IaaS	Infrastructure as a service
PaaS	Platform as a Service
GD	Gradient Decent
SCGD	Scaled Conjugate Gradient Decent
VM	Virtual Machine
CPU	Central Processing Unit
MSE	Mean Square Error
RBNN	Recurrent Backpropagation Neural Network
SLA	Service Level Agreement
LSTM	Long Short-Term Memory
N-LSTM	Novel Long Short-Term Memory
RVLRBNN	Random Variable Learning Rate Based Back Propagation Neural network

CHAPTER ONE

1. INTRODUCTION

1.1 Background

Cloud computing technology was emerged was in 996 for distributing computing resource in cloud services' providing and becoming one of the dominant technologies due to all services are migrating to cloud-based services. Workload prediction is the process of predicting how system workloads will vary in the future from current traffic flow to the system. In cloud services providing, multiple machines with different location internal computing resources allows sharing a single physical instance of resource or an application to multiple users. Those multiple machines use different resource of cloud data center. The main purpose of future workload prediction is to provide reduction time window for deployment of cloud physical servers and virtual machine creation and their allocation. In addition to provides services with high quality, to stay in market, to know customer interest, to provide services as per as services level agreement (SLA), and to avoid over providing which leads to loss of product and under providing which lead to loss of potential customer, cloud services provider should have a prediction model to predict their future resource management based on their past traffic flow. Nowadays, due to the increasing demand for cloud-based services, accurate workload prediction becoming big challenge for liner and traditional prediction model while efficient computational resource utilization and resource management in cloud datacenter become an important task. Virtualization is a process by which the users of a cloud service share the data present in the cloud, which can be application software, storage, and platform.

The concept of virtualization was emerged in early 1970 to solve problem with cloud computing technologies. In virtualization technology, the server and the software application which required by the services provider maintain by the third party and in this, the cloud provider please some amount to the third party. Cloud platform has a dynamic resource usage as it shared among large number of users. Accurately prediction of incoming workload to cloud virtual machine in such dynamic environment very challenges task due to nonlinear and dynamic behavior of incoming workload to cloud services. The task is difficult for traditional prediction model, it is not easy to capture strong dependence, pattern, and relation among workload to cloud. In traditional prediction model, users request to virtual machine regular, and they have linear pattern. So, the task easy for linear prediction models to learn easily and predict future workload within a given accuracy level for cloud virtual machine.

Previously, many prediction models had used to solve problem of prediction where the historical records of cloud data or traffic flow have nonlinear and inconsistent behavior. But, some of them concluded that clear forecasting of future resources' requirement require model which can capture dynamic characters and create pattern among different request to delivery services to customer as per as agreement. Proper prediction needs a model which can capture dynamic nature of virtual machine request and provide more accurate prediction for future resource requirement. Many researches had done to overcome the problem of workload prediction at cloud data center from previous historical log data. Some of them done by machine learning algorithms due to machine learning based prediction can handle none linearity properties of workload.

Machine learning techniques are none linear model and can yield accurate prediction result. Machine learning, ensemble mechanism is renowned for improving the prediction accuracy which uses learners rather than single learner. Neural network is one of the machines learning method to solve the stated issues in workload prediction with high performance. Neural Network is one of the machine learning built to synthesize complex none linearity which can give the better accuracy level. Back propagation supervised learning algorithm and most common, and widely used learning algorithm. Back propagation is a gradient descent-based algorithm and called traditional back propagation algorithm. The algorithm calculates the output of the network model and reduces the Mean Square Error between the actual output and the desired output through adjusting weights accordingly [4,5]. The problem with traditional back propagation neural network slow learning process, increases number of neurons to increase prediction accuracy, limited to the learning rate given by the user, it trapped into a local minimum and, when learning rate is set too high, the algorithm may oscillate and become unstable and when learning rate is too small, the algorithm will take too long to converge.

There are many variations of traditional BPNN algorithm, those are, Scaled Conjugate algorithm [32] which provide the method in which error function decrease very rapidly and faster training algorithm comparison to back propagation one. Conjugate gradient with Fletcher -reeves updates [33], provide method to calculate next weight and bias in through norm square of both previous and current gradient, Conjugate Gradient with Polak-Ribiere update is the recent version of Conjugate Gradient proposed by Polak- Ribiere [34] and Prop algorithm [35] which provide faster training rate and time and have capability to escape from local minima.

Levenberg Merquadrt algorithm is a combination of two first order algorithm steepest-descent and gauss-newton method. Levenberg Merquadrt algorithm is a second derivative algorithm. However, steepest-descent adjusts the weights by using the gradient descent method. If the learning rate is set too high, the algorithm may oscillate and become unstable and if the learning rate is too small, the algorithm will take too long to converge. The major drawbacks of traditional BPNN are the slow learning process and used user selected learning rate. The Levenberg Marquardt uses gradient to update weight at every learning iteration. So proper selection of weight is necessary for next leering and weight update and accuracy yield at end of learning since gradient are very sensitive to the initial weight network. With the randomly selected initial weights and biases, the neural network cannot accurately estimate the required output. The weights are update based on the error values corresponding to the complete error over the training set.

Levenberg Merquadrt algorithm uses back propagation learning uses gradient to update weight which is similar with gradient descent algorithm. Gradient descent algorithm sensitive to initial weight and bias. In other word, gradient descent is prone to giving local maximum and minimum values. Levenberg Marquardt algorithm always assumed independent of its initial weight and used default weight initialization. Lack of proper weight initialization leads to the algorithm reduction in accuracy level of prediction, raises dead number of neurons and increasing convergence rate as nonlinear workload provided to it and due to stated issues and, gradient disappearance sometimes happens. The disappearance of the gradient will cause the weights of the layers in neural network almost unchanged while training model, still close to the initial weights after many iterations.

To overcome stated problem with Levenberg-Merquadrt algorithm and to yield better accuracy for workload prediction of virtual machines in cloud computing this work designed and presented weight initialization based random weight assignment to neural network to avoid dead number of neurons and sensitivity of gradient to initial weight to yield better prediction accuracy for cloud virtual machine for future resource requirement. The work done by enhancing Levenberg Marquardt Back propagation algorithm based proper initial weight initialization method to improve prediction accuracy level for future resource requirement and allocation of cloud virtual machines. The presented work reduced gradient sensitivity of initial weight, avoid death of neuron and disappearance of the gradient for neural network training model and improve prediction accuracy and with acceptable convergence rate.

Generally, motivation to this paper is the existence of fast-growing technology of cloud-based services where all services are migrating due to many advantages of cloud technology. But, the nature of incoming workload to cloud virtual machine had been becoming nonlinear from time to time. So, the nonlinearity and inconsistent behavior of cloud workload need accuracy level in prediction for efficient resource management, to reduce time for resource allocation and deployment of virtual machine, providing high quality of services and used for resource utilization from cloud services provider point of view. Predicting future demand of cloud computing virtual machine is very important task for cloud services management, for providing time window creating, allocation and scheduling of resource at cloud data center. The main purposes of predicting future requirement of cloud resources are, for dynamically provisioning through avoiding under provisioning and over provisioning resources which reduces the risk of wasting resources during non-peak hours and reduce missing potential customers respectively [2].

1.2 Statement of the Problem

Cloud computing is the dynamic environment that provide on demand services over internet as you go model. Utilizing cloud computing resources remain very challenge due to inconsistent behavior of incoming workload to cloud services. So, it is challenges to predict this by using traditional prior prediction algorithms. Improving prediction accuracy cloud virtual machine workload remain extremely challenges due to the cloud virtual machine workload fluctuate dramatically at small timescale [21]. Workload on cloud virtual machine fluctuate dynamically, and it is nonlinear due to that it is difficult for traditional prediction model to create pattern and find hidden attribute within different resources request and virtual machine requirement.

As cloud services provider the challenges are, how to improve prediction of future resource requirement with incontinent cloud workload behavior, having mechanism to avoid over provisioning of cloud resource which leads to wasting services during non-peak hours and under provisioning of them which leads to missing of potential customers to provide services as per as SLA made with customer and for existence in the market and to be competent enough, having prediction model with more accuracy level also help services provider to minimize time schedule and allocate resources to incoming request based on previous historical data and having prediction model which allow fast allocation of virtual machine to incoming workload to cloud services. Among stated problem with cloud workload prediction for future resource requirement and management, some of them listed as below,

- ✓ Less prediction accuracy as incoming workload to cloud becoming nonlinear
- ✓ Increase dead numbers of neurons to enhance prediction accuracy for existing machine learning
- ✓ Default weight initialization used which leads to gradient disappearance as well higher convergency rate as input volume to learning model increases

1.3 Objectives of the Study

1.3.1 General Objective

The general objective of this thesis work is, to develop and implement proper weight initialization based Levenberg–Marquardt algorithm for workload prediction cloud virtual machines to improve prediction accuracy level.

1.3.2 Specific Objective

In general, this thesis work has the following specific objectives: -To identify existing problem with Levenberg–Marquardt algorithm and to enhance prediction accuracy for workload prediction to cloud virtual machine.

- ✓ Reviewing different existing work done for workload prediction.

- ✓ Developing model to overcome Levenberg–Marquardt algorithm.
- ✓ Selecting proper methodology, implementation, and simulation tools.
- ✓ Developing algorithm for identified stated issue.
- ✓ Implementing developed method to overcome raised gap
- ✓ Simulating developed prediction model
- ✓ Writing the simulation result of new improved algorithm.
- ✓ Evaluating performance of newly developed model with existing one.
- ✓ Comparison of obtained result with existing model.

1.4 Methodology

To develop method that overcome raised problem in our work and obtain the desired result, the following methods used while doing this thesis stating from finding knowledge gap to implementing and simulating proposed result.

1.4.1 Quantitative Method

From the beginning, quantitative research method used to analysis and utilize numeric data using different statical techniques to answer question like; how much, how many, how etc. The method also explaining an issue or phenomenon through gathering data in numerical form. Since, our study tries to show improvement, contract among prediction model in term of numeric value, different workload measurement value with time interval and put their result in numeric values, our paper used quantitative method for calculating them for model performance improvement, to plot of input datasets, to record training iteration, updating next weight, and averaging samples rate of virtual machine.

1.4.2 Literature Review

Different literatures searched and reviewed from journals and different Conferences on workload prediction for cloud virtual machines. Among those done by machine learning algorithm and other prior algorithms. The reviewed papers used to know what had done in the area, to select algorithm among existing, to prioritize among prediction model, to get knowledge gap and problem in the areas, how to improve them and to develop method and algorithm for improvement of performance for workload prediction accuracy of virtual machine.

1.4.3 Simulation Tool

To do this thesis, different designing, implementation, and simulation tools used to show achievement of our objectives and improvement to enhance workload prediction. The detail description of those tools stated as below.

MATLAB_R2016a: is a high-performance programming language used for numerical computing. MATLAB allow matrix manipulation, plotting of graph and function, data implementation of algorithm and numbers of toolboxes, built in neural network algorithm and simulation environment. In short, MATLAB used to design neural network model for proposed system, for graphical representation of cloud none linear workload including memory intensive workload and CPU intensive workload, for writing code for improvement method to existing method and for implementation and performance evaluation of new system.

Smart Draw 2019: is application software which provides different drop and drag toolboxes and arrow for designing flow chart, models, and it is easy to work with. Smart Draw 2019 had used to design Neural network architecture and back propagation learning and to design flow chart for enhanced Levenberg Marquardt back propagation algorithm with weight initialization mechanism.

Wondershare EdrawMax: is an application software contain libraries like basic shape, arrow, and line with different resizing method to draw flow chart, data diagram and node for design. The Wondershare EdrawMax had used to draw flow chart and system architecture model. It is good application and user-friendly application

1.5 Scope and Limitation of the Study

The scope of the paper is to design and implement the workload prediction model based proper weight initialization to get reasonable accuracy level of cloud virtual machine future resource requirement. The proposed prediction model can predict future workload based on historical data of users request to virtual machines in which CPU and Memory intensive workload used for evaluating performance proposed model. The model used to solve problem of future resource prediction with nonlinear and irregular time intervals at request level. The range of cloud incoming work prediction accuracy level measured by" Mean Square Error" which is default performance measurement function (PerformFcn) for 'LM' algorithm. Intensive workload of two performance measurement metrics historical data within different time stamp provided to prediction model and the response of model recorded which measured in 'MSE' and other parameters for performance measurement considered as default value. Created prediction model add proper weight initialization mechanism to input dataset to provide prediction accuracy. Random weight initialization used for generating weight and assignment of them to two neural network nodes to after they checked for similarity of weight assignment at different node.

Generally, as any work have their own limitation, our work also has the below limitations:

- ✓ Due to sample dataset used for training, numbers input dataset with their time interval taken from previous work (reference work) the model does not test for long term prediction.
- ✓ The work also tested for the prediction of only two major component of computing resource those are CPU and Memory intensive workload due to they have golden contribution for building virtual machine.
- ✓ The work does not check level of pattern among incoming workload to cloud services to provide services based on their pattern to cluster services.
- ✓ While enhancing performance of algorithm only accuracy considered which measured in 'mse' other parameters of algorithm used with default value.

1.6 Application of The Result

The result of this paper is applicable in cloud computing technology where incoming workload of cloud virtual machine wanted to be predicting based on history of virtual machine requests trace for future resource requirement for services management and scheduling. Improving accuracy level on workload prediction enable cloud services management launching their services efficiently and effectively by avoiding over provisioning and under provisioning, resource scheduling and configuring their resources in a manageable way. Generally, the result of thesis will use for cloud services those can be predict in terms of virtual machine log file obtained from cloud data center.

1.7 Organization of the Paper

The rest of this paper had organized as follows: Method. Chapter 2, discusses Literature review on introduction to Cloud Computing Technology, virtualization technology, Machine Learning Model, Neural Network Back Propagation, Levenberg–Marquardt Back-Propagation Algorithm and Weight Initialization techniques. Chapter 3, discusses on different Related Work which had done on area based Neural Network Back propagation algorithms and those initiated by weight initialization method Chapter 4: discusses on proposed design model, design module and architecture, system flow chart, tools used for designing and pseudocode. Chapter 5: discusses on implementation and result, training data set by graph and implementation tools. Chapter 6: discusses conclusion and future work of paper

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Introduction

Cloud computing is the fast-growing technology which had been using widely for distributing and acquiring the required resources through Internet. It refers to the delivery of infrastructure components, software, storage, and services over the Internet based on user demand with pay-as-you-go subscription Cloud computing is the dynamic environment that provide on demand services over internet as you go model [7,14]. Cloud computing is internet-based technology where virtual shared servers provide software, infrastructure, platform, devices and other resources and hosting to customers on a pay-as-you-use [3, 2]. Cloud computing is the on-demand availability computer resources especially computing storage and power, without direct active management by the user. The term cloud-based services generally used to describe data centers available to many users over the internet large clouds, predominant today, often have functions distributed over multiple locations from central servers.

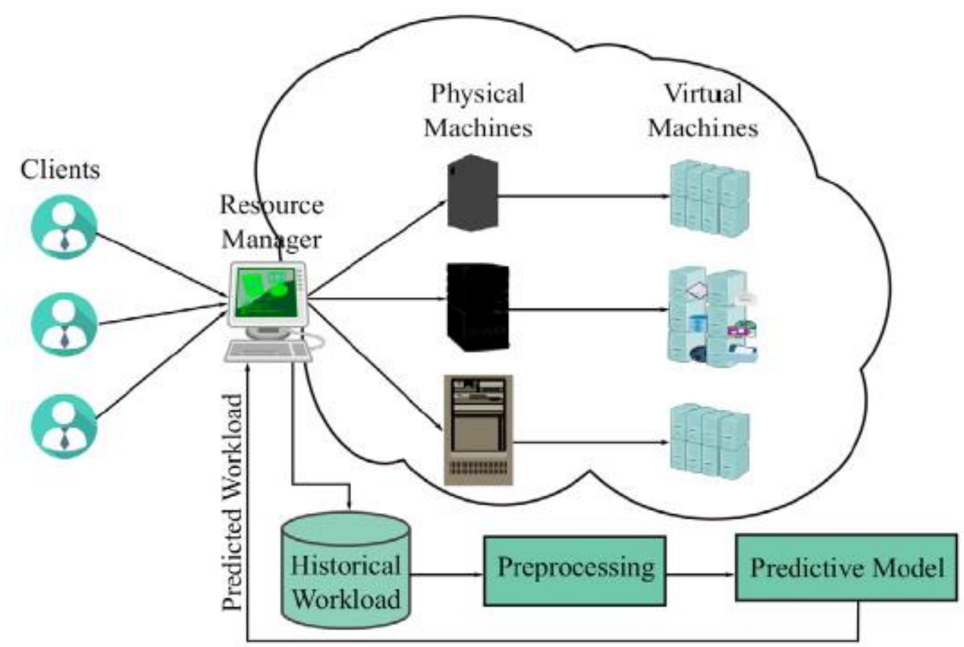

Figure 1 Cloud Computing Architecture [2]

2.2 **Cloud Computing Model**

Cloud computing models come in three different models: Those models are, Software as a Service (SaaS), Infrastructure as a Service (IaaS) and platform as a Service (PaaS) [19,25]. When each of the cloud models have their own set of benefits that could serve the needs of various businesses.

Software as a service, provides a facility to the user to use the software from anywhere with the help of an internet connection. It is also known as software on demand. The remote access is possible because of service providers, host applications and their associated data at their location.

Infrastructure as a Service, is basically a virtual provision of computing resources over the cloud. An IaaS cloud provider can give you the entire range of computing infrastructures such as storage, servers, networking hardware alongside maintenance and support.

Platform as a Service, is essentially a cloud base where you can develop, test, and organize the different applications for your business. Implementing PaaS simplifies the process of enterprise software development. The virtual runtime environment provided by PaaS gives a favorable space for developing and testing applications.

Qazi Zia Ullah, [3] defined different cloud computing deployment models of cloud computing like public cloud, community cloud, hybrid cloud and private clouds.

Public Cloud Model is where the company serves the infrastructure to the customer on a commercial basis. This type of service model enable customer to develop and deploy the application with minimum financial outlay.

Community Cloud Model, those companies having the same interest and work can share the same cloud, and it can be doing with the help of community cloud. The initial business startup saved, as the setup established.

Hybrid cloud model, provide an easy way to move the application to move from one cloud to another and, it is a combination of public and private cloud which supports the requirement to handle data in an organization.

Private cloud model is cloud company maintains the management, deployment, and operation of the cloud. The operation can be in-house or with a third party. While delivering services for its customer, cloud computing had been facing with different challenges like security & privacy, interoperability & portability reliable and flexible, cost, downtime, lack of resources and management of multi-cloud environment

2.3 Cloud Virtual Machine

Virtualization in Cloud Computing is the mechanism of making a virtual platform of server operating system and storage devices [7]. The method helps the user by providing multiple machines at the same time, it also allows sharing a single physical instance of a resource or an application to multiple users. Cloud Virtualization also manage the workload by transforming traditional computing and make it more scalable, economical, and efficient. One of the important features of virtualization is that it allows sharing of applications to multiple customers and companies. [3,14] stated virtualization in cloud computing is a process in which the user of cloud shares the data present in the cloud which can be application software etc.

2.4 Machine Learning

Machine learning is a branch of intelligence approach and the intersection of computer science [20]. Its statistics used for data analysis and automates analytical model building based on the idea that systems can learn from data. ML is capable to identify patterns and make decisions with minimal human intervention [19]. The approach taken from pattern recognition and the theory in which computers can learn without programmed to perform specific tasks, researchers interested in artificial intelligence wanted to see if computers could learn from data. There are different types of machine learning model like, supervised, unsupervised and semi-supervised. Machine learning evolved from the study of pattern recognition and aims at giving computers the ability to decide problems without human intervention [9,19]. I's explicitly programmed. Neural Network nets to synthesize complex none linearity [4]. ANN is one of the which can give the better accuracy level almost every time has made neural nets the most popular solution for almost all machine learning related difficulties. Simply choosing and specifying the weight initialization by randomization function reduces the risk of the training progress slowed to the point of impracticality. ANN has three major learning paradigms. Machine learning, basically, categorized into three groups, unsupervised machine learning, supervised machine learning and reinforcement machine learning [12,20]. It is also a model highly improves the accuracy of, robustness, and prediction and the generalization ability of the conventional time series forecasting tools.

2.4.1 Supervised learning

Supervised learning is the type of machine learning in which learning rule provided with a set of training data of proper network behavior. As inputs applies to the network, the network outputs are comparing to the targets [20,25]. In supervised learning, the computer taught by example. It learns from past data and applies the learning to present data to predict future events. In this case, both input and desired output data provide help to the prediction of future events.

2.4.2 Unsupervised learning

Unsupervised learning is a machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses datasets. The most common unsupervised learning method is cluster analysis. The method which had been using for exploratory data analysis to find hidden patterns or grouping in data. The type of learning process where weights and biases have modified in response to network inputs only. There are no target outputs available. Most of these algorithms perform clustering operation. They learn to categorize the input patterns into a finite number of classes [20]. Machine learning categorized to three main categories.

2.4.3 Reinforcement Learning

Reinforcement learning it is a type of learning rule like supervised learning, except that, instead of providing with the correct output for each network input, the algorithm only gives a grade. The grade is a measure of the network performance over some sequence of inputs [24]. Reinforcement learning differs from supervised learning in not needing labelled input output pairs presented and in not needing suboptimal actions explicitly corrected

2.5 Artificial Neural Network

Artificial neural networks the component of artificial intelligence that meant to simulate the functioning of a human brain. Processing units make up ANN, which in turn consist of inputs and outputs. The inputs are what the ANN learns from to produce the desired output. Artificial neural networks are computing systems vaguely inspired by the biological neural that constitute animal brain [4]. Such systems ‘learn’ to perform tasks by considering examples, generally without programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have manually labeled [18]. A neural network is a massively parallel distributed processor made up of simple processing units that have a natural tendency for storing experiential knowledge and making it available for us. ANN is a type of artificial intelligence technique that mimics the behavior of the human brain. ANN can model linear and non-linear systems without the need to make assumptions implicitly as in most traditional statistical approaches [37].

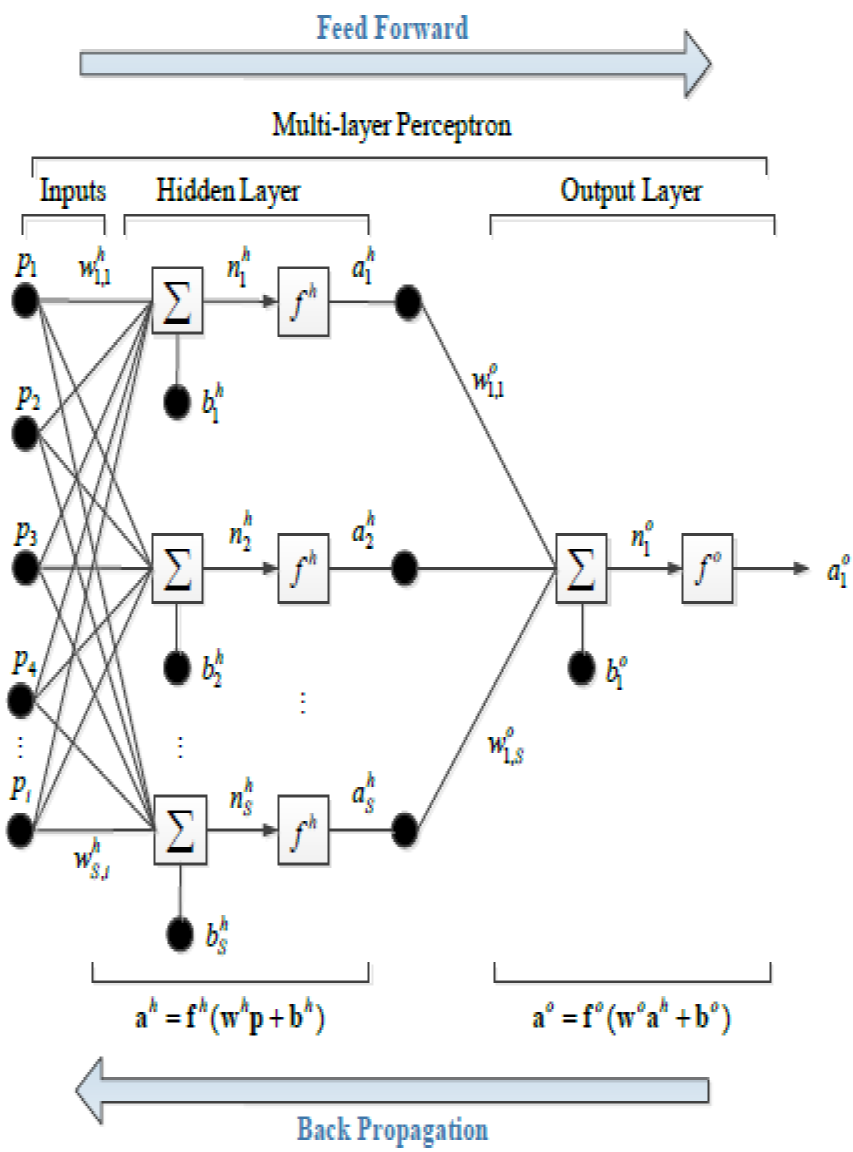

Figure 2 Artificial Neural Network Model [37]

2.6 Components of Neural Network

Artificial neural networks are comprised of many elements. Those elements are building block of neural network and some of them discussed in below

2.6.1 Neuron

ANN are composed of artificial neural which retain the biological concept of neuron [18]. Neurons are component which receive input, combine the input with their internal state and an optional threshold using an activation function, and produce output using an output function. A neuron is a mathematical function that model the functioning of a biological neuron. Typically, neuron computes the weighted average of its input, and this sum passes through a nonlinear function often called the activation function.

2.6.2 Weight

The network consists of connections, each connection providing the output of one neuron as an input to another neuron [19]. Every connection assigned a weight that represents its relative importance. A weight represents the strength of the connection between units. If the weight from node to node2 has greater magnitude, it means that neuron1 has greater influence over neuron.

2.6.3 Bias

Bias is simply a constant value of those added to the product of inputs and weights. Bias utilized to offset the result. The bias use to shift the result of activation function towards the positive or negative side. It is an additional parameter in the neural network, which used to adjust the output along with the weighted sum of the inputs to the neuron. Therefore, bias is a constant which helps the model in a way that it can fit best for the given data.

2.6.4 Activation Function

In a neural network, activation functions are a critical component. Activation functions determine the output of a learning model, its accuracy, and the computational efficiency of training a model, and which can make or break a large-scale neural network. Activation functions also have a major effect on the neural network’s ability to converge and the convergence speed, or in some cases, activation functions might prevent neural networks from converging in the first place.

2.7 Weight Initialization Method

Weight initialization is an important consideration in the design of a neural network model. Several weights' initialization for multilayer perceptron, an initial forward pass-through neural network performed using an initialization set weight. Using statics obtained from these set, weights initialized to neural network. Different method to initialize weight [22]. The role of weight initialization is, to prevent layer activation outputs from exploding or vanishing during a forward pass in the neural network. If either occurs, loss gradients will either be too large or too small to flow backwards, and the network will take longer to converge. Proper initialization method is one of the most significant prerequisites for fast convergence of feed-forward neural networks like high order and multilayer perceptions. The aim of weight initialization is to determine the optimal value of the initial weight in vector matrix, variance among weight.

2.7.1 Zero Weight Initialization

The type of weight initialization for neural network where weight of each neuron is set to zero [14]. Initialization weight of all neurons with zero values have no purpose for training since all neuron produce identical output whatever activation function used

2.7.2 Random Weight Initialization

New weight initialization for sigmoid function [26]. The method developed based on random weight initialization method. Random weight initialization weights randomly selected from range [L, -L] where L is max value and -L stands for small values, and it is the type of weight initialization method of neural network where weights provided to network model with mean value of zero and standard deviation values one. The method can yield better accuracy if numbers of hidden nodes have random distributed weights. The accuracy level of random weight initialization depends on numbers of hidden neuron.

2.7.3 Xavier Weight Initialization

The type of weight initialization method where variance of given neural network node depend on previous node. Variance of node reduced by 1/n when n is numbers of neural node [7].

2.7.4 He-at-al Weight Initialization

The method like Xavier weight initialization, with only the difference being a factor of Xavier initialization multiplied by two [31].

2.8 Back Propagation Learning

Back propagation supervised learning algorithm and most common, and widely used learning algorithm. Back propagation is a gradient descent-based algorithm and called traditional back propagation algorithm. The algorithm calculates the output of the network model and reduces the Mean Square Error (MSE) between the actual output and the desired output through adjusting weights accordingly [4]. Back propagation supervised learning algorithm and most common, and widely used learning algorithm. Back propagation is a gradient descent-based algorithm and called traditional back propagation algorithm. The algorithm calculates the output of the network model and reduces the Mean Square Error between the actual output and the desired output through adjusting weights accordingly.

When learning rate is set too high, the algorithm may oscillate and become unstable. However, if the learning rate is too small, the algorithm will take too long to converge. The problem with traditional back propagation neural network slow learning process, increases number of neurons to increase prediction accuracy, limited to the learning rate given by the user, it trapped into a local minimum and, when learning rate is set too high, the algorithm may oscillate and become unstable and when learning rate is too small, the algorithm will take too long to converge.

2.8.1 Levenberg Marquardt Algorithm

LM back propagation algorithm is one of machine learning algorithm which developed in the early 1960s to solve nonlinear least squares problems [14,22]. Least squares issues are coming in the context of fitting a parameterized function to a set of measured data points by reducing the sum of the squares of the errors between the input and output of the function. Nonlinear least squares methods iteratively minimize the sum of the squares of the errors between the function and the measured data points through a sequence of updates to parameter values. Levenberg-Merquadrt algorithm has the fastest convergence and train network 10-100 times faster than BPNN [4].

2.9 Applications of Neural Network

Due to some of its wonderful properties, ANN have many applications. Those properties are ability to take in a lot of inputs, process them to infer hidden as well as complex, non-linear relationships. ANN are playing a big role in different areas. Character recognition like handwriting has a lot of applications in fraud detection, e.g., bank fraud and even national security assessments. Image recognition is an ever-growing field with widespread applications from facial recognition in social media, cancer detention in medicine to satellite imagery processing for agricultural and defense usage. The research on ANN now has paved the way for deep neural networks that forms the basis of "deep learning" and which has now opened all the exciting and transformational innovations in computer vision, speech recognition, natural language processing, famous examples being self-driving cars.

2.9.1 Forecasting

Neural network forecasting required extensively in everyday business decisions. Examples -sales, financial allocation between products, capacity utilization in economic and monetary policy, in finance and stock market. More often, forecasting problems are complex, for example, predicting stock prices is a complex problem with a lot of underlying factors some known, some unseen. Traditional forecasting models throw up limitations in terms of considering these complex, non-linear relationships [16]. ANN, applied in the right way, can provide a robust alternative, given its ability to model and extract unseen features and relationships. Also, unlike these traditional models, ANN does not impose any restriction on input and residual distributions.

2.9.2 Speech Recognition

Speech recognition system converts the speech signals and decodes them to text or some form of meaning. We can say it is a direct example of applications in virtual assistants or chatbots. Nowadays, Google smart home, Alexa, Siri, Google assistance [30].

2.9.3 Text Classification

Text classification is an essential part in many applications, such as web searching, information filtering, language identification, readability assessment, and sentiment analysis. Neural networks actively used for this task.

2.9.4 Clustering

Clustering is one of the applications of neural network and machine learning technique that involves the grouping of data points [29]. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

2.9.5 Classification

Classification also another neural network application and it is the final stage of the pattern recognition. This is the stage where an automated system declares that the inputted object belongs to a particular category [29]. There are many classification methods in the field. Classification method designs are based on the following concepts

2.9.6 Feature Extraction

Feature extraction is the process of studying and deriving useful information from the filtered input patterns [9]. The derived information may be general features, which are, evaluated to ease further processing. For example, in image recognition, the extracted features will contain information about gray shade, texture, shape or context of the image. This is the main information used in image processing. The methods of feature extraction and the extracted features are application dependent.

2.9.7 Pattern Recognition

The act of recognition divided into two broad categories: recognizing concrete items and recognizing abstract items the recognition of concrete items involves the recognition of spatial and temporal items: - Examples of spatial items are fingerprints, weather maps, pictures, and physical objects. Examples of temporal items are waveforms and signatures.

CHAPTER THREE

3. RELATED WORK

3.1 Introduction

Several researches had been conducting with motivation of workload prediction of cloud environment. The reason workload prediction required is now a day, all services are migrating to cloud technology for different cloud benefits like economical due to its paid as used, easy for business startup, reliability, manageability, data centralization and scalability. So, cloud services provider had been using different prediction model. Workload on cloud services is fluctuating within time interval. Different papers had done to maintain stated problems. Those related works are range from neural network prediction algorithm to those initiated by weight initialization-based algorithm and discussed below.

[10,13] prediction of cloud workload for cloud resource utilization presented. The author tried to overcome the importance of cloud resource utilization for efficient and effective cloud services management. The work also provided prediction at per task and per resource level mechanism. The evaluated performance of proposed model reduces prediction error. The accurate prediction of cloud resource utilization needs experimental research with detail resource threshold. [1] workload of cloud datacenter proposed based Neural Network for time series cloud workload. The developed method can provide better accuracy level linear issue.

[39], the cloud virtual machine placement for live migration-based host load detection done by Markov prediction model. The work presented mechanism for host overload and under load for incoming workload to cloud machine for efficient resource unitization and cloud virtual machine creation and migration. The purpose of method to avoid immediate virtual migration based on future perdition based on previous history data. The work simulated in Cloud Sim and show significant in violence of service level agreement which cause loss of customers in cloud technology. In [36] the authors presented workload prediction scheme for cloud virtual machine allocation based on differential evolution. Basically, prediction method based on evolution are very important hence they, can capture hidden properties of training inputs and memorize different point in training at each layer of training model. The developed model training Google

request dataset and showed better in significant error reduction performance over others back propagation algorithm, but performance of algorithm slows down as input data size increases.

The workload of cloud services based on neural network back propagation algorithm and [28,37]. The short-term workload forecasting of cloud based on improved neural network back propagation and two stage neural networks. The authors presented two stage neural network method for cloud resource clustering and predicting for effective resource utilization. The work cluster cloud resources history into three classes: - over, normal and under, and then apply prediction algorithm. Workload prediction based on neural network back propagation the effective virtual machine strategies and significant error forecasting error is better in back propagation algorithm. The problem with this forecasting model unable to capture long term dependence among different time interval workloads and among different iteration gradient which used for updating next weight and bias provided to training model

[21] Container load prediction of cloud applications developed with deep neural network. The work provides better mechanism of cloud load prediction and, provided method to select important metrics for training which reduce dimensionality training parameters used in neural network training for future container load prediction. What should appreciate from this work a mechanism to select essential attributes of load for training machine learning because the work reduced dimensionality of training parameters for training. Reduction in parameters dimensionality also used to enhance memory requirement for training algorithms. But they had evaluated the performance of algorithm evaluated against prior and traditional time series-based algorithm and cloud workload now a day becomes nonlinear problem which need accuracy in prediction with this dynamic environment.

In [23], the work presented based on convolutional neural networks for forecasting short-term data center network traffic load. The model developed to overcome the issue of traditional prediction algorithm that they failed to obtain accurate prediction. The developed model based on convolutional network can handle nonlinear traffic in cloud environment, but the model can handle short term traffic flow. The work done by convolutional neural network based can improve prediction accuracy level, but they can only predict for short term traffic flow to cloud datacenter.

In [6] author developed workload forecasting for smart cloud based on Random Variable Length Rate Back Propagation Neural Network RVLBPNN. The paper developed method based on weight initialization technique and improved prediction accuracy on captured cloud workload where CPU and Memory intensive workload have taken as training and testing dataset to predict future resource's requirement. This work used as reference to our thesis also. The work designed and implemented in MATLAB 7.14 language in which provide built in simulation environment. Performance of algorithm measured in Mean Square Error to show prediction accuracy level. Proposed system evaluated against HMM and NBC algorithms showed better performance in average Mean Square Error. The work improved accuracy level by using proper weight initialization for smart cloud dataset.

[38], proposed and implemented cloud cluster workload prediction based on recurrent neural network. The paper used differential transformation and orthogonal experimental design method for normalization and finding most influential parameters in training and applied weight initialization selected to activation function for training and selecting sample data for training neural network. The paper used CPU and Memory intensive workload to training model and the method showed better prediction accuracy level for short term prediction. Recurrent Neural Network is one of back propagation training model and used when accuracy of prediction required. RNN based prediction model, hidden layer neurons have memory to save state of last input to create pattern among different inputs. In RNN prediction model, hidden neurons are having memory of point from previous iteration to create relationship among different interval of workload for future estimation.

The big problem with standard RNN is they cannot provide required prediction accuracy for long term prediction, it increases number of hidden neurons, it saves state of previous input which need high memory as numbers of input data volume increases.[37], deployed workload prediction of virtual machine based on improved neural network proposed and developed. The model overcome issue of LSTM and RNN by providing the mechanism to train virtual machine historical workload data with irregular time intervals of workload history. The method also captured log history and provided long term relationship existence among virtual machines at different workload. The model used historical data of users request to virtual machine for training and evaluated the result against RNN and LSTM to show accuracy level. The author of paper called developed model called N-LSTM and finally, their result showed more accurate prediction level than other existing prediction model.

Neural network back propagation algorithm overcome problems of linear prediction algorithm through training large dataset in nonlinear workload environment. But still, some of BNN algorithm need improvement in training parameters for accuracy of prediction, especially in cloud computing environment. Since BNN uses gradient for next weight and bias adjustment, weight initialization is one of the mechanisms to overcome issue of sensitivity to initial weight and disappearance of gradient. Many works have been to overcome difficulty in nonlinear condition based on different weight initialization mechanism. Those reviewed and discussed as below.

In [11,15] different weight initialization method for neural network presented. The work discussed the techniques of initial weight selection from different dataset and compared different weight initialization method with their perspective activation function. The work also showed how initial weight selection affect result in training iteration and their dependence to action function used in BNN.[17] The paper presented approaches to initialize weight to network model for improving weight initialization of Relu activation function. The paper tried to show importance of weight initialization and their relation to activation function. Prone of the model is it takes weight from training dataset which can improve accuracy but require high memory for storing trained dataset. The weight initialization influenced by activation function used for training neural network.

[16] the work proposed accurate weight initialization method for multilayer feed forward. The work addressed weight initialization method influence like neural network algorithm training parameters speed of convergence, probability of convergence and the generalization are those altered. [8,27] the work discussed on different initialization of weight in neural network and compared the performance of different neural network algorithm based on initial weight

selection. The approaches' framework for weight initialization and different dataset from which weights initialized. The work avoids the problem neural network overfitting and vanishing problem. Generally, for some related work we summarized as below conclusion part to show their contribution and drawbacks.

Mainly, as discussed in reviewed related work, machine learning based cloud workload prediction model developed to overcome limitation of time series-based prediction mode. They have better performance in cloud incoming workload prediction in nonlinear and incontinence cloud incoming workload environment. Nowadays, every application is moving to cloud-based services due to its advantage and incoming workload to cloud virtual machines are dynamic and varies within time interval and becoming nonlinear so due to these behaviors accurate prediction of cloud services is complex task.

The problem of neural network is a gradient descent-based and use user selected learning rate parameter. As learning rate is set too high, the algorithm may oscillate and become unstable. Neural network back propagation algorithm also contributes in providing attribute selection and reduction input data to training network model. They also investigated a method to Slashdot, which is unpredictable flash crowd workloads and not handled by traditional prediction algorithm.

The other component of related work, prediction algorithm based on weight initialization. They tried to solve problem of some neural network -BNN prediction algorithm like speed of convergence, accuracy in prediction as training dataset increase through different weight initialization techniques. Selection of weight initialization for training network model depend on training algorithm and activation function used. Generally, BNN with fixed learning rate cannot extract nonlinear relationship among sample of large dataset. BNN converges slowly as sample of training data increase. Improved BNN prediction can extract hidden relationship among cloud work. Levenberg Marquardt back propagation algorithm is one of the learning algorithms developed to overcome problem of traditional BNN and improved BNN in nonlinear problem and provide better accuracy and convergence rate as sample data increase. The algorithm also needs small memory requirement for recording training dataset. But performance of the algorithm can improve by proper weight initialization method to initial input data to training network model.

Many researches have done on cloud workload prediction by using different machine learning algorithm for linear and nonlinear complex issues. Most of the papers also done by Levenberg Marquardt algorithm and another traditional algorithm. Performance of the algorithm was good in dynamic workload cloud environment to predict future resources' requirement, but prediction accuracy can improve. Levenberg Marquardt algorithm update weight by gradient descent algorithm and sensitive to initial weight and bias.in other word, gradient descent is prone to giving local maximum and minimum values.

Training of any neural network is sensitive to initial weight choice and bias. Weight initialization contributes as a significant factor on the final quality of a network as well as its convergence rate [11]. Proposed system designed and implemented in MATLAB R2016a and performance of proposed system evaluated against existing algorithm to improve prediction accuracy on smart cloud. Traditional prediction model cannot handle complex observation or where workload to cloud services is nonlinear, and those models can work well in some kinds of workload by assuming that the future will follow the same pattern as in the past.

The main contribution of this paper is to design and present improved workload prediction mechanism by enhancing Levenberg Marquardt algorithm on cloud datacenter which based enhanced Levenberg–Marquart back propagation algorithm. The thesis prepared model based proper weight initialization mechanism to avoid gradient sensitivity of initial weight and to provide better prediction accuracy, and convergence rate.

Table 3. 2 Summary of Related Work

No	Article Name	Autor’s Name	Year of publication	Contribution of the work	Drawback of the work

1	Markov Prediction Model for Host Load Detection and VM Placement in Live Migration	Suben Bani Melhem, Anjali Agarwal.	December, 2017.	Provide mechanism for host overload and underload for incoming workload to cloud virtual machine for efficient resource unitization and cloud virtual machine creation and migration.	It takes long computational time and memory to train large input dataset while learning.
2	RVLBVNN: A Workload Furcating Model for Smart Cloud Computing	John Panneerselvam and Yan Wu.	December, 2016.	Provide resource utilization mechanism by evaluating the model performance CPU and Memory intensive workload input data to predict future resources requirement.	Random weight initialization and variable learning rate are being using for training, improving prediction accuracy and good convergency.
3	Workload Prediction of Cloud Cluster Using Recurrent Neural Network.	Weishan Zhang, Bo Li, Dehai and Zhaho	October ,2019.	Recurrent Neural Network based model show better prediction accuracy level for short term prediction and having input state to hidden neuron.	Lack of providing required prediction accuracy for long term prediction and long-term dependence among input data
4	An Efficient Workload Prediction in Cloud Computing Using Two Stage Neural Network.	K. Dinesh Kumer and E. Umameheswari	May,2019.	The work tried to solve problem with prior neural network algorithms by providing long-term dependency method among input data to learning model	Lack of handling workload of irregular interval and loss of prediction accuracy for large input data volume.
5	Short-Term Load Forecasting of Virtual Machine Based on Improved Neural Network.	Xudong Lu and Hui Lu	December,2016.	Provide method to overcome problem of both LSTM and RNN and Handle irregularity behavior workload.	Lack of proper weight initialization and increases dead numbers of neuron to enhance prediction accuracy.
6	Fisher: an efficient container load prediction with deep neural network in clouds.	Xuehai Tang, Quiyang Liu, and Yangchen Dong	September,2019.	Provide method to select important metrics for training and reduce dimensionality training parameters used in neural network training for future container load prediction in cloud.	Performance of algorithm evaluated against prior and traditional time series-based algorithm

CHAPTER FOUR

4. PROPOSED SYSTEM

4.1 Overview

In this chapter, the paper present overview of proposed system model including, deployment architecture, flow chart for proposed model with their pseudocodes, design consideration parameters and to enhance Levenberg Marquardt algorithm back propagation algorithm based on proper weight initialization accuracy of prediction for cloud virtual machine workload.

To overcome the stated problem, the proposed system used proper weight initialization that based on random initialization of weight to input dataset weight (i, j). The method based proper input data normalization to have small data value [0,1] for machine learning and check variance in data point vector matrix. The existing Levenberg Marquardt back propagation algorithm uses default weight initializing method and check only if input data point in input vector matrix has empty weight only not variance among dataset for training. The model avoids sensitivity of gradient to initial point of weight and dead neuron, through providing a method to check variance among weight to neurons. Paper present prediction mechanism based improved Levenberg Marquardt algorithm for more accurate prediction level with acceptable values of performance measuring parameters like convergence rate, gradient, and numbers epoch to provide prediction of future workload in virtual machine by using historical data of CPU and Memory intensive workload for more resources' utilization from cloud services management and providing point of view.

4.2 Proposed Architecture

The proposed architecture to show architecture of Neural Network model based on our selected LM algorithm. This architecture shows the improvement of weight initialization to ANN model by omitting default weight selection, which lead performance of algorithm to stated problem in our previous problem of statement. As indicated in both architecture and flow chart of our suggested system, in our work as input provided to model for learning variance of each weight in the eight matrices must check to avoid sensitivity of gradient to initial weight since the gradient always used for updating of next weight to minimize error and for convergence.

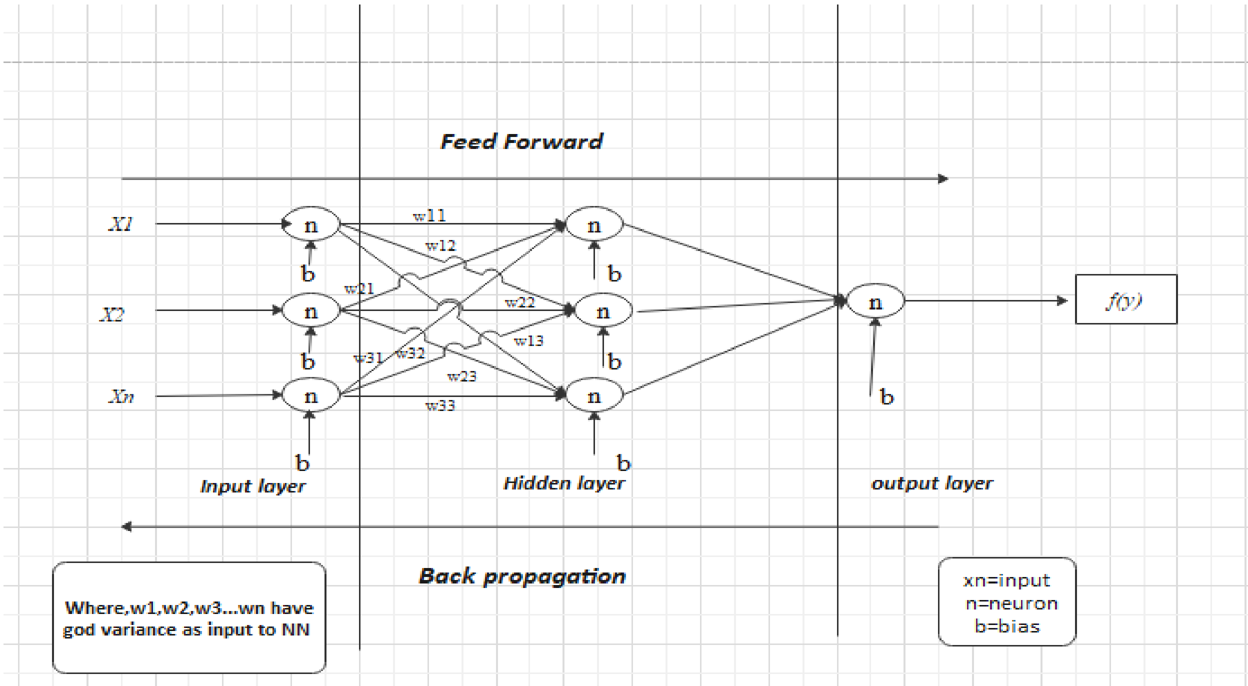

Figure 3 Proposed Architecture of NN Model

4.3 Deployment module

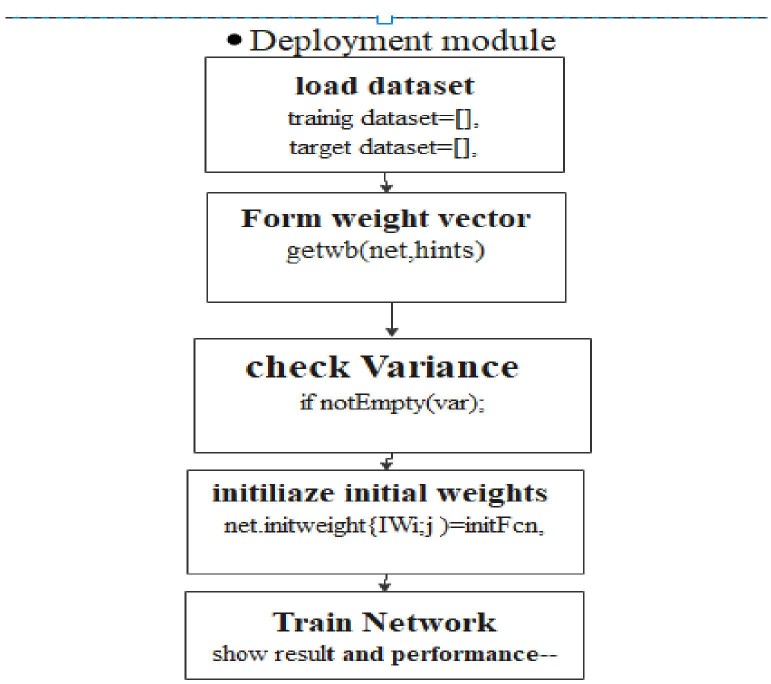

Figure 4 Proposed System Deployment Module

4.4 Proposed Flow Chart

The thesis developed flow chart of proposed which based on initial weight initialization method to solve stated problems ‘LM’ back propagation algorithm to show prediction accuracy level of incoming workload to cloud virtual machines by considering internal intensive CPU and Memory workload for better resource utilization. Improved flow chart of ‘LM’ algorithm showed as below

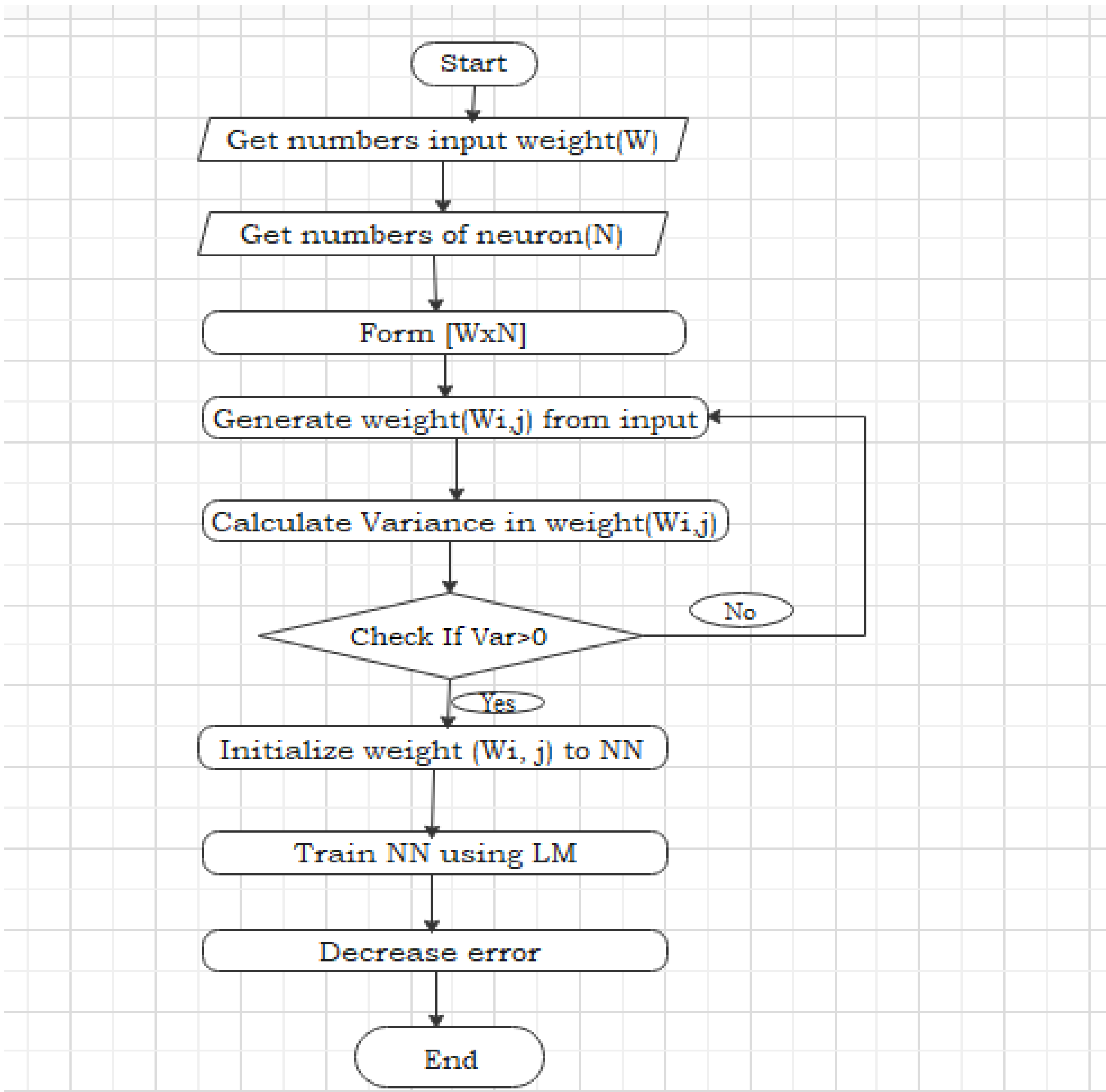

Figure 5 Proposed Flow Chart

4.5 Pseudocode

Proposed system to improve prediction accuracy level for workload of cloud virtual machine based on enhanced Levenberg Marquardt backpropagation algorithm with proper initial weight selection method algorithm has developed pseudo code. Those pseud codes contain code to initialize input weight to NN model, code to get new weight randomly in from provided input in vector weight, code to get new weight randomly in from provided input in vector weight, code to train NN model with 'mse' default parameter, to remove zero-Var inputs and to remove unused inputs from NN.

```
%To initialize input weight to NN model*

function settings = configure weight(inputs)

settings. input Size = size(inputs,1);

end

function w = initialize_input_weight (net, i, j, config)

w = norm (rands (net. Layers{i}. size, config. input Size));

end

function w = initialize_layer_weight (net, i, j, config)

w = norm (rands (net. Layers{i}. size, config. input Size));

end

%To get new weight randomly from provided input vector
weight*

function x = new_value_from_rows_cols (rows, cols)

x = norm (rands (rows, cols));

end
```

```
function x = new_value_from_rows_range (rows, range)

x = norm (rands (rows, size(range,1)))

%To replace new weight with zero var weight in weight vector*

ww = w;

Var= zeros (s, R);

w (: ind) = ww

%To train NN model with ‘mse’ default parameter*

function net = format Net(net)

if is empty (net. PerformFcn)

warning (message ('nnet: train: EmptyPerformanceFixed'));

net. PerformFcn = 'mse';

net. performParam = mse('defaultParam');

end

if is empty (nnstring. first_match (net. PerformFcn, {'sse','mse'}))

warning (message ('nnet: train: NonSqrErrorFixed'));

net. PerformFcn = 'mse';

net. performParam = mse('defaultParam');

end

% To remove Zero-Var Inputs*

delInputs = false (1, net. numInputs);
```

```
for i=net. numInputs: -1:1
if (net. Inputs{i}. var == 0) || (net. Inputs{i}. processed Size == 0)
net = nn_delete_input (net, i);
delInputs(i) = true;
change = true;
end
end
% To remove unused inputs*
kept Inputs = find(~delInputs);
for i = net. numInputs: -1:1**** for unused input
if ~any (net. input connect (, i))
net = nn_delete_input (net, i);
delInputs(keptInputs(i)) = true;
keptInputs(i) = [];
change = true;
end
end
delInputs = find(delInputs);
%To remove zero-sized input weights*
for i=1:net. numLayers
```

```
for j=1:net. numInputs

if net. input connect (i, j)

if is empty (net. input Weights {i, j}. delays)

net. input connect (i, j) = false;

net. input Weights {i, j} = [];

net. IW {i, j} = [];

change = true;

end
```

CHAPTER FIVE

5. IMPLEMENTATION AND EVALUATION

5.1 Overview

In this chapter, the work presented the deployment overview of enhanced Levenberg Marquardt back propagation algorithm for workload prediction in cloud virtual machines at CPU and Memory intensive workload level to ensure more resource utilization from cloud services provider point of view. The work ensures prediction accuracy level for reducing time create virtual machine, allocation, migration and creating time window for physical machine installation. Software tools used to design, implement, obtaining training data set, validation and testing data set and parameters used for performance evaluation of proposed system.

The proposed model to enhance cloud workload prediction model for cloud virtual machine based on Levenberg Marquardt back propagation algorithm based on proper weight initialization mechanism to avoid sensitivity in initial weight selection, reducing dead neurons for better prediction accuracy and acceptable convergence rate in nonlinear incoming workload to cloud environment. The proposed model developed and implemented in MATLAB R2016a programming language. Historical data of users request to virtual machine used for prediction of future virtual machine requirement. Virtual machine run independently but share physical machine resource. Those front and major resource components mainly used for virtual machine creation and allocation are CPU and Memory historical dataset used for training the model. In our work, Memory and CPU intensive workload of cloud virtual machine parameters used for performance evaluation and their result showed as below.

5.2 Implementation Tool

MATLAB_R2016a: used to implement suggested system**.** MATLAB_ R2016a is a high-performance programming language used for numerical computing. It provides different machine learning algorithm in its toolbox. Levenberg Marquardt back propagation algorithm is the one as default training algorithm found in toolbox with URL: C:\Program-Files\MATLAB\R2016a\toolbox\nnet\nnet\nntrain/trainlm. MATLAB allow matrix manipulation, plotting of graph and function, data implementation of algorithm and numbers of toolboxes, built in neural network algorithm and simulation environment. In short, MATLAB used to design neural network model for proposed system, for graphical representation of cloud none linear workload including memory intensive workload and CPU intensive workload. It also allows performance evaluation for prediction, called Mean Square Error

5.3 Dataset

The datasets are collection of 28 days google cluster workload history with one hour interval of cloud virtual machine data usage consisting of 4,609,320 tasks comparison of CPU intensive workload and Memory intensive workload and both [6]. To show the prediction accuracy level of suggested system, the model-trained input dataset of log file in terms of their memory and CPU workload data for the given time stamp for predicting future workload estimation cloud datacenter. The historical values of Memory and CPU requirement workload retrieved from, and their workload recorded by time stamp. Number of time stamp and their intensive workload recorded metrics used for performance evaluation of proposed model. The input data record to prediction model has time stamp of 10minute, 20 minutes and 30 minutes with the same time stamp for target data which is the same with our reference paper. The input data samples trained in MATLAB R2016a. The collected historical dataset workload trace used for training have shown as below diagram with nonlinear and inconsistent behavior as shown below

5.3.1 Input Dataset

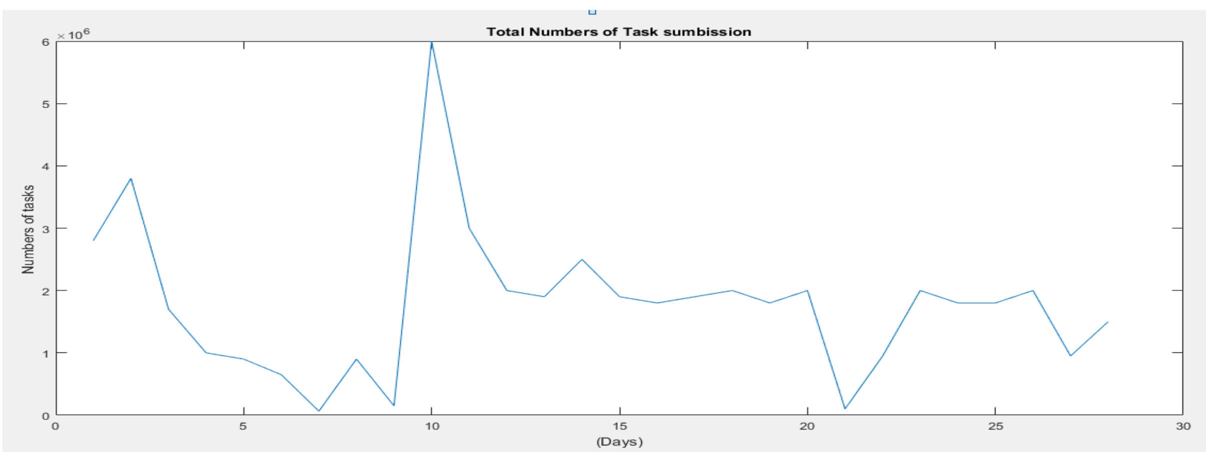

Figure 6 Trace log of Total Numbers of Workload to Google Cluster

The proposed prediction model trained with 10 virtual machines samples historical CPU intensive workload record with different time stamp input data and Memory intensive workload. The work contracts the prediction with successive 10 sample data. Different volume of input data for different time stamp record provided to prediction model and numeric response of model recorded for sample data. We created the model with three layers neural network and same numbers input, hidden and output layers neuron and activation function at each layer are the same with our reference paper. From collected dataset of Google cluster trace workload with 28 days which contain both Memory and CPU intensive workload, The collected dataset link taken from our reference paper and each of their intensive workload contain 10 successive samples with different workload provided to developed neural network model. Their response of network to each sample are record.

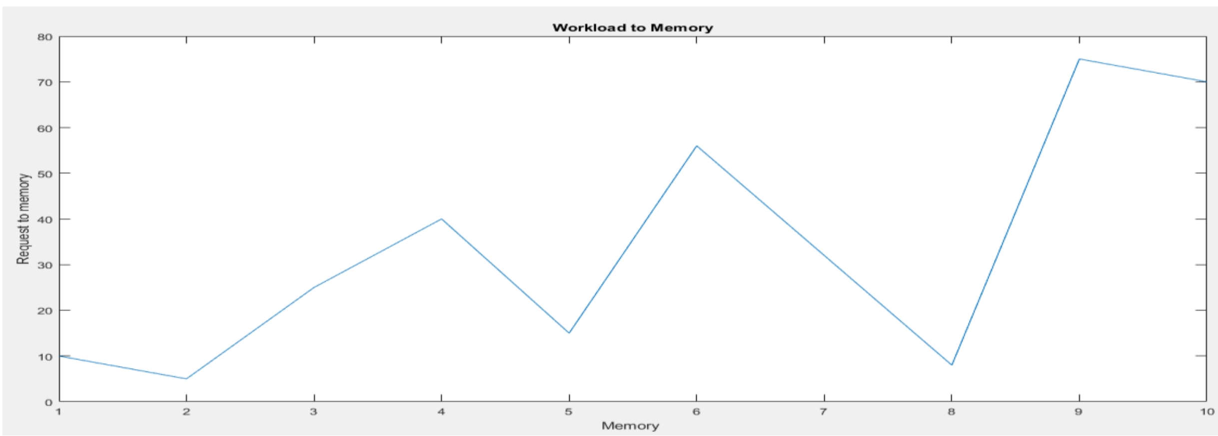

Figure 7 Workload of Virtual Machine in Terms of Memory

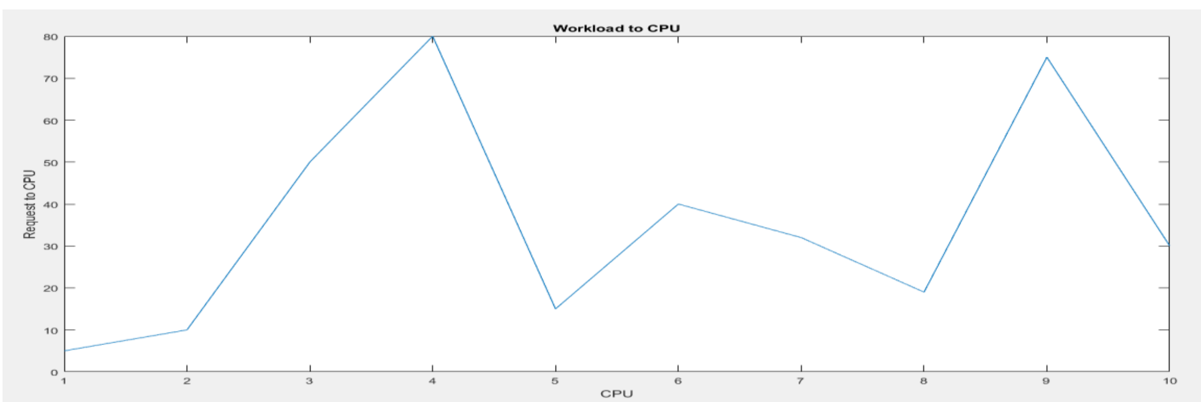

Figure 8 Workload of Virtual Machine in terms of CPU

5.3.2 Target Dataset

From collected cloud virtual machine historical data in terms of intensive CPU and Memory workload for 10 sample data, each successive 10 sample data with different volume with different time stamp used as target data to train model with the same volume of data with our reference work

5.3.3 Test Dataset

The back propagation model always dumped input sample dataset to input dataset and test datasets (80% training dataset and 20% test datasets) sample as default rule of any machine learning algorithm.

5.4 Performance Evaluation

The proposed enhanced Levenberg Marquardt back propagation algorithm to improve prediction level on cloud virtual machine cluster used the below parameters for measuring the performance and evaluating it with existing 'LM' back propagation neural network algorithm to show performance of as proposed. Performance evaluation parameters defined as default (default 'MSE') values in Levenberg Marquardt back propagation algorithm and used to monitor newly suggested algorithm which measure the accuracy of algorithm in terms of response to the model as input provided to it.

5.4.1 Prediction Accuracy

Prediction accuracy of most machine learning algorithm called 'performance'**.** Prediction accuracy, expressed as the correlation between the predicted and actual values. It is performance evaluation parameters measured in terms of' MSE' between the actual workload on cloud virtual machine and the desired workload on cloud cluster which labeled as target in our work. The objective of MSE it to make the difference between input data point target either zero or near zero. Parameters like convergence rate and numbers of neuron affect performance of neural network workload prediction accuracy measured in terms of Mean Square Error between actual (target workload minus desired workload, and mathematically calculated as below

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(yi - y\hat{}i)\ 2$$

Were, MSE = Mean Square Error

n= is numbers of data point

$yi = observed\ value$

$y^{\wedge i} = predicted\ value$

Hence, the objective of neural network training is to reduce or to make zero the difference between target and desired (network output) workload, minimum value of MSE is always preferable. Since, network training data have more sample, the average of MSE of training result taken to get performance of our model.

Convergence rate called epoch. Epoch is one of the main criteria to evaluate performance of machine learning algorithm. As numbers of input dataset increase in machine learning algorithm, convergence rate of some algorithm decrease. These parameters indicate how fast machine learning train the given dataset to obtain training goal. It measured in numeric values and small values always selected as volume of input data to prediction model increases.IN our case, default value of algorithm.

Numbers of Neural network node used for training model network play a great role. Because, hidden node put map between input and output node in back propagation. Some works tried to increase prediction accuracy level by through increasing numbers of hidden layer neuron. To evaluate performance of proposed model, numbers of network layer, input nodes, hidden nodes, outer nodes, and activation function used at each layer of training model are similar with the reference paper used for our paper.

Specifically, the learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0. The learning rate controls how quickly the model is adapted to the problem and the default values of algorithm used without change.

5.5 Simulation

MATLAB R2016a is a high-performance programming language used for numerical computing. It provides different machine learning algorithm in its toolbox. Levenberg Marquardt back propagation algorithm is the one as default training algorithm found in toolbox with URL: C:\ProgramFiles\MATLAB\R2016a\toolbox\nnet\nnet\nntrain/trainlm. MATLAB allow matrix manipulation, plotting of graph and function, data implementation of algorithm and numbers of toolboxes, built in neural network algorithm and simulation environment. Provide built in simulation environment. The simulation used 10 sample historical workload input data recorded by time stamp of different volume with the 10 next successive sample data of time interval provided to model and the computed numeric numbers of training performance recorded for all 10-sample training dataset in terms of both Memory and CPU intensive workload. The simulation network model created by calling function nntool and numbers of input, layers, adaptive function, numbers output and activation function are the same with our reference paper except numbers of neurons different in our scenario to avoid dead numbers neuron with proper initial weight initialization. For remaining parameters of algorithm, default values used.

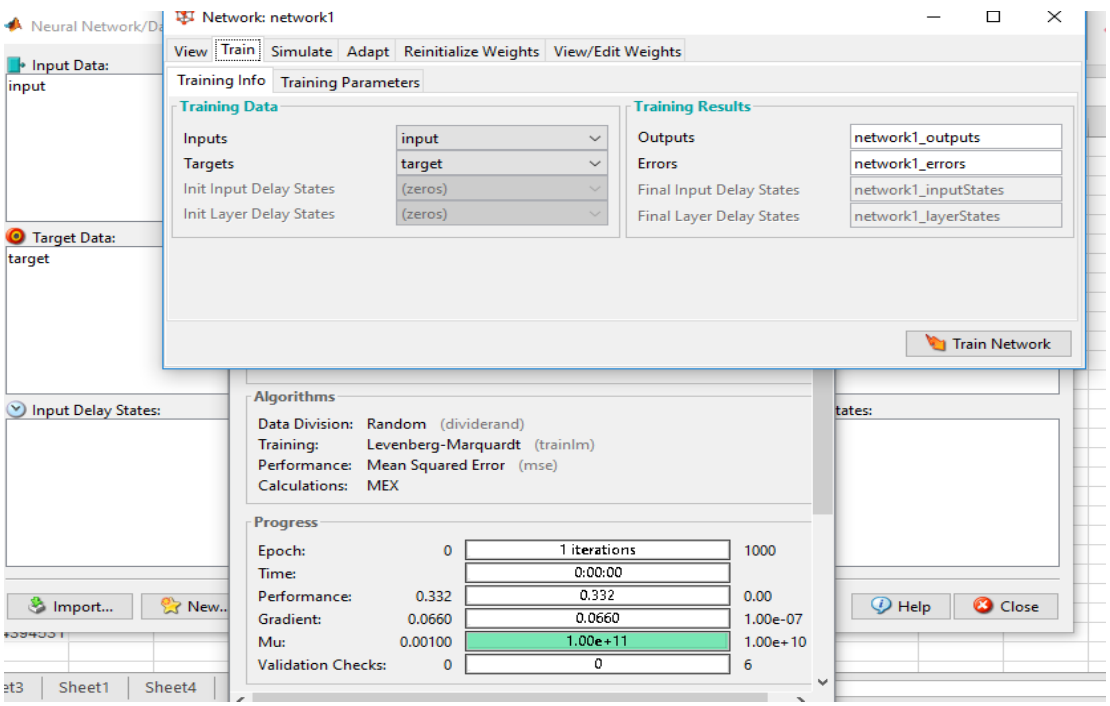

Figure 9 Network Simulation Model

5.6 Result

Proposed enhanced Levenberg Marquardt back propagation algorithm implemented and simulated MATLAB R2016a programming language which provide built 'LM' algorithm based on proper initial weight initialization method. Sample data of workloads in terms of CPU and Memory intensive workloads provided to network training model. The model trained with 10 sample data of both CPU and Memory intensive workload and performance of each sample recorded. The average accuracy rate of workload estimation in 10 sample data plotted in bar graph figure 10 and figure 11. And, the proposed model shows better in terms of time stamp to create virtual machines as shown in figure 12 in this result average prediction accuracy of ten virtual machines taken against time window taken to create and assign them to workload delivered to cloud datacenter.

To show performance of suggested system, the result compared to the reference paper which done by RNN based prediction model. Performance of values measured Mean-Square-Error [1]. Since 'MSE' measure the difference between input datasets and target value, small numeric values selected for performance evaluation.

To show performance of our work graphically for samples of trained data, average prediction accuracy percentage (y-axis) and the response of developed training model (x-axis) used to sketch the graphs. Response of model to workload and its prediction accuracy level for both CPU and Memory, interpreted and discussed as below to show improvement.

5.6.1 CPU Workload Perdition

As indicated in Figure 9, after providing 10 sample input data with three different time stamp record which leads to different in volume cloud workloads of virtual machine in terms of CPU intensive workload data to proposed system enhanced Levenberg–Marquardt back propagation neural network algorithm and the result of samples recorded. Performance of network measured in' MSE' and average value of all samples measured for level accuracy rate.

The numbers of back propagation algorithm x(axis) plotted against their average predict ion accuracy rate of training samples data y(axis). The average accuracy percentage estimation is about

61.78 and 62.45% for our developed model and reference respectively for CPU intensive workload prediction accuracy. The numeric values in percentage and graph are evident to show performance of proposed model to provide better prediction accuracy in different nonlinear workload as input datasets to training model increases with minimum average of 'MSE'.

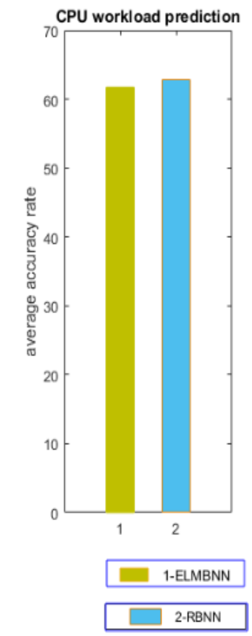

Figure 10 CPU Intensive Workload Prediction Result

5.6.2 Memory Workload Prediction

As indicated in Figure 10, the same scenario repeated for Memory workload prediction as the same sample data provided to training model, their responses recorded. Performance of training model measured in' MSE' as previous. The average of numeric values for overall trained sample taken for level of accuracy rate. The numbers' prediction model x(axis) against average prediction accuracy rate y(axis) plotted to show improvement gained over existing one.

The average accuracy percentage estimation is about 61.80%and 62.50% for our proposed and reference model respectively for Memory intensive workload prediction accuracy. The numeric values in percentage and graph are evident to show performance of proposed model provide better prediction accuracy in nonlinear workload with minimum average of MSE over existing method.

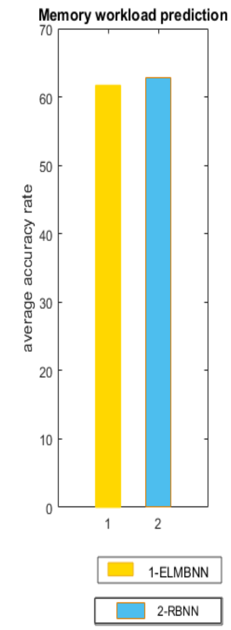

Figure 11 Memory Workload Prediction Result

5.6.3 Interpretation and Discussion

The simulation result obtained for sample workload data for both CPU and Memory workload of cloud virtual machine represented in graph are evident for proposed system based proper weight initialization show better performance than existing 'LM' algorithm. Suggested and developed model maintains better convergence rate as volume of input data increases and avoid dead hidden neuron since weight initialized with good variance to neuron to omit disappearance of gradient which used to update next weight to decreases error in a learning. So, the model which based on proper weight initial weight initialization can predict incoming workload to cloud virtual machine with better accuracy level with acceptable convergence rate than other prediction model in nonlinear and inconsistent workload environment. The work provides better resource utilization than other prediction model in advance to resource allocation, scheduling, and management.

CHAPTER SIX

6. CONCLUSION AND RECOMMENDATION

6.1 Conclusion

As showed in result of both Memory and CPU workload cloud virtual machine log data, the average 'Mean Square Error' result of sample data based on proper initial weight initialization method named as enhanced Levenberg–Marquardt back propagation algorithm labeled as ELMBNN in our obtained result, perform better prediction accuracy level in nonlinear and inconsistent cloud virtual machine workload environment. Also, as prediction accuracy increase, time elapsed to create a virtual machine to assign to datacenter request decreases. Nowadays, and near future, world will be moving to cloud-based service. So, cloud services provider needs workload prediction with reasonable accuracy level for future resource requirement, dynamic provisioning, scheduling of resources, to reduce time for creating and allocating of virtual machine and virtual machine migration to meet customer need. One of the better mechanisms to overcome those issues' provider having good prediction model. They always demand for prediction model those can extract map between different workload and train high volume datasets to get hidden attributes for long term prediction. Smart prediction algorithms require a method to handle between user behavioral change to cloud dataset overtime. So, machine learning method is one of mechanism to overcome the problem. Proposed system based on Levenberg–Marquardt back propagation algorithm is the better prediction algorithm in nonlinear workload environment and dynamic behavior of incoming workload to virtual machine due to it is the second order algorithm and fastest among existing family of back propagation learning

6.2 Recommendation

Currently, all services are migrating to cloud technology. Incoming workload to cloud virtual machine becoming more nonlinear and inconsistent. Unless proper perdition of future resources' requirement to cloud data center, it will be more challenges for cloud services provider to meet customer need with minim cost of operation. BNN algorithm based predictive model are very important by handling long-term dependence and training large data set for future prediction. LM back propagation algorithm is most powerful, capable to handle nonlinear problem, fast and provide more accuracy than existing algorithm. So, in near future it is better to enhance parameters like method of weight updating and transfer function with variance checking in hidden neurons of this algorithm to use for more dynamic incoming workload to cloud services.

Reference

[1]. Carlos Vazquez, Ram Krishnan, and Eugene John, “Time Series Forecasting of Cloud Data Center Workloads for Dynamic Resource Provisioning”, IEEE, Journal of Wireless Mobile Networks, Ubiquitous Computing, And Dependable Applications, Volume: 6, Number, 2016.

[2]. Ali Yadavar Nikravesh, Samuel A. Ajila* And Chung-Horng Lung,” An Autonomic Prediction Suite for Cloud Resource Provisioning”, Journal of Cloud Computing: Advances, Systems and Applications (2017) 6:3

[3]. Qazi Zia Ullah, Shahzad Hassan, And Gul Muhammad Khan3,” Adaptive Resource Utilization Prediction System for Infrastructure as A Service Cloud”, Hindawi Computational Intelligence and Neuroscience Volume 2017.

[4]. Chien-Sheng Chen,” Artificial Neural Network for Location Estimation in Wireless Communication Systems”, ISSN 1424-8220, 12, 2798-2817, Sensors 2012.

[5]. B.V.V.S Prasad, Guntur, and Andhra Pradesh, “Predicting Future Resource Requirement for Efficient Resource Management in Cloud”, International Journal of Computer Applications (0975 – 8887) Volume 101– No.15, September 2014.

[6]. Yao Lu,1 John Panneerselvam,2 Lu Liu, And Yan Wu, “RVLBPNN: A Workload Forecasting Model for Smart Cloud Computing”, ISSN, Computational Intelligence and Neuroscience, Volume 2016.

[7]. S. Sridevi, Dr. Jeevaa Katiravan Assistant Professor, Associate Professor, “A Study on Load Prediction Methods for Optimal Resource Allocation in The Cloud Environment”, ISSN: 2249-2593, International Journal of Computer & Organization Trends (IJCOT) – Volume 7 Issue 4 –July to Augest,2017.

[8]. Abdirashid Salad Nur, Nor Haizan Mohd Radzi and Asharaf Osumman Ibrahim, “Neural Network Weight Optimization: A Review”, ResearchGate, Telkomnika Indonesia Journal of Electrical Engineering. 12 No 9, September, Pp. 6897-6902.

[9]. Rathinapriya Vasu, E. Iniya Nehru and G. Ramakrishnan, “Load Forecasting for Optimal Resource Allocation in Cloud Computing Using Neural Method”, ISSN, Middle-East Journal of Scientific Research 24 (6): 1995-2002, 2016.

[10]. Kritwara Rattanaopas Picahye Tadayya “Adaptive Workload Prediction for Cloud-Based Server Infrastructure” E-ISSN, Journal of Telecomminiation, Electronic and Computer Engineering. 9 No. 2-4, May 2014.

[11]. Priyesh Patel, Meet Nandu, Purva Ruat” Initialization of Weights in Neural Networks”, ISSN:2455-2631, Departement of Information Technology, Vol. 3, November 2018.

[12]. Amir Mosavi, Abudulla Bahmani,” Energy Consumption Prediction Using Machine Learning; Review” IEEE, Department of Computer Science, Norwegian University Od Science Technology,11 March ,2019.

[13]. Michael Borkowski, Stefan Shulte and Christophi Hochriener,” Prediction of Cloud Computing Resource Utilization” ResearchGate, Conference Paper, December,2016.

[14]. Jemes Dillinger, “Weight Initialization in Neural Networks”, Journal Paper, April,2019.

[15]. Mercedes Fernandez-Redondo, Carlos Hernandez-Espinosa” Weight Initialization Method for Multilayer Feedward1”, ISBN, ISANN2001-Preccedding European Symposium on Artificial Neural Network, Belgium,25-27 April 2001.

[16]. Denis Erdogmus, Oscar Fontenla-Remero, Jose C. Principe, Amparo Alonso-Betansos, Enrique Castillo, Robert Jenssen “Accurate Initialization of Neural Network Weight by Propagation of Desired Response”, IEEE Ixpoler, Conference Paper, Agust,2003.

[17]. Diego Agurrie Olac Fuentes,” Improving Weight Initialization of Relu and Output Layers” ResearchGate, September, 2019.

[18]. Radhakrishnan Ayyapazham And Kavitha Veluatham" Proficient Decision Making on Virtual Machines Creation in IaaS Cloud Environment" The International Arab Journal of Information Technology 14, No 3, May 2017.

[19]. Vanish Jatana," Dive into Machine Learning, ResearchGate, Book, November,2018.

[20]. Mohssen Mohammed, Mohammed Badruddin, Khan, Eihab Bashier Mohammed Bashier "Introduction to Machine Learning", ResearchGate, July 2016.

[21]. Xuehai Tang, Quiyang Liu, Yangchen Dong, Zighon Han, Zhiyuan Zhang" Fisher: An Efficient Container Load Prediction with Deep Neural Network in Elemental Conf with Parallel and Distributed Processing with Application, Ubiquitous Computing and Communications ,2018.

[22]. Vikram Kumer, Akhil Guitar Sanjeev Aggarwal, "Artificial Neural Network-Based Load Forecasting Using Levenberg Marquardt Algorithm", International Journal of Advanced Research in Computer Science and Software Guitar 4, January 2014.

[23]. Alberto Monobrow Ordozgoiti, Sandara Gomez-Canaval "Forecasting Short-Term Data Center Network Traffic Load With Convolutional Neural Networks",

[24]. Sivasankari Bhagavathperumal, Madhu Goyal" Dynamic Provisioning of Cloud Resource Based on Workload Prediction", Center of Artificial Intelligence, Factuality of Engineering and Information Technology, NSW 2007.

[25]. Mad.Tanjil Sarker and Sanjida Noor," Basic Application and Study of Artificial Neural Network", International Journal of and Research Hub, ISSN:2494-3122, Volume 4 Issue 4 April,2017.

[26]. Sartaj Singh Sodi, Pravin Chandra and Sharad Tanwar" A New Weight Initialization Method for Sigmoid Feedforward Artificial Neural Networks ", Soda Conference on Neural Network (JCNN), July 6-11,2014.

[27]. Young-Man Kwon, Young-Woo Kwon, Dong Kewun Chung and Myung Jae Lim" The Comparison of Performance According to Initialization Method of Deep

Neural Network for Malware Dataset", Lim "The Journal of Innovative Technology and Exploring Engineering (EJITEE), Volume-8 Issue-4S2, March,2019.

[28]. K. Dinesh Kumer, E. Umameheswari," EWTPNN: An Efficient Workload Prediction in Cloud Computing Using Two Stage Neural Networks" Elsevier, International Conference on Recent Trend on Advanced Computing (ICRTAC),2019.

[29]. John Peter Jesan" The Neural Approach to Pattern Recognition", ACM Digital Library, Ubiquity, Volume Issue April, 2004

[30]. Google, Google Cluster Data V1, Https://Github.Com/Google /Cluster-Data/Blob/Master/Clusterdata2011_2.Md.

[31]. Min Sun and Yju Wang," The Conjugate Gradient Method for Solving the Generalized Sylvester Matrix Equations", Springer, Journal of Applied Mathematics and Computing,12 October 2018.

[32]. Handif D Sherali and Osman Ulular," Conjugate Gradient Method Quasi-Newton Updated with Inexact Line Searches", Elsevier, Journal of Mathematical Analysis and Application, Volume 150, August 1990, Page 357-377.

[33]. Dr Abbas Al-Bayati, Dr Khalil K. Abbo, and Ibrahim A Saleh" Conjugate Gradient Back-Propagation with Modified Polak-Ribiere Update for Training Feedforward Neural Network", Reaserchgate, The Fourth Scientific Conference of Colleges of Computer Science and Mathematics,09 July,2019, Pp [164-173].

[34]. Chiritsan Igel And Micheal Huksen," Improving the Rprop Learning Algorithm ", Reaserchgate, Proceeding of The Second International Symposium on Neural Computation, 21 May 2014, PP.115-121.

[35]. Xavier Glorot And Yoshua Bengio," and Understanding the Difficult of Deep Feedforward Neural Network "Researchgate, Article on Journal of Machine Learning Research, May,2014.

[36]. Jitendra Kumar And Ashutosh Kumar Singh," Cloud Resource Prediction Using Differential Evolution Based on Learning", IEEE, International Conference on Smart Communication and Computing (ICSCC),2019

[37]. Wei Guo, Wei Ge, Xudong Lu and Hui Lu," Short-Term Load Forecasting of Virtual Machine Based on Improved Neural Network", IEEE, Journal Paper, Volume .4 ,2016.

[38]. Weishan Zhang, Bo Li, Dehai, Zhaho, Faming Gong, and Qingau Lu," Workload Prediction of Cloud Cluster Using Recurrent Neural Network", IEEE, Conference Paper, China, October,2017.

[39]. Suben Bani Melhem, Anjali Agarwal, Nishith Goel and Marzia Zamal," Markov Prediction Model for Host Load Detection and VM Placement in Live Zamal," Markov, Digital Object Identifier 10.1109, China, Volume 6, December 19,2017.

Appendix

```
[worker. perf, worker.vperf,worker.tperf,worker.je,worker.jj,worker.gradient] =
calcLib.perfsJEJJ(calcNet);
if calcLib.isMainWorker
    worker.epoch = 0;
    worker.startTime = clock;
    worker.param = archNet.trainParam;
    worker.originalNet = calcNet;
    [worker.best,worker.val_fail] = nntraining.validation_start(calcNet,worker.perf,worker.vperf);
    worker.WB = calcLib.getwb(calcNet);
    worker.lengthWB = length(worker.WB);
    worker.ii = sparse(1:worker.lengthWB,1:worker.lengthWB,ones(1,worker.lengthWB));
    worker.mu = worker.param.mu;
    worker.tr = nnet.trainingRecord.start(tr,worker.param.goal,...
        {'epoch','time','perf','vperf','tperf','mu','gradient','val_fail'});
    worker.status = ...
        [ ...
        nntraining.status('Epoch','iterations','linear','discrete',0,worker.param.epochs,0), ...
        nntraining.status('Time','seconds','linear','discrete',0,worker.param.time,0), ...
        nntraining.status('Performance','','log','continu-
ous',worker.perf,worker.param.goal,worker.perf) ...
        nntraining.status('Gradient','','log','continuous',worker.gradi-
ent,worker.param.min_grad,worker.gradient) ...
        nntraining.status('Mu','','log','continuous',worker.mu,worker.param.mu_max,worker.mu) ...
        nntraining.status('Validation Checks','','linear','discrete',0,worker.param.max_fail,0) ...
        ];
end
end
Stopping Criteria
current_time = etime(clock,worker.startTime);
[userStop,userCancel] =  nntraintool('check');
if userStop
```

```
        worker.tr.stop = message('nnet:trainingStop:UserStop');
        calcNet = worker.best.net;
      elseif userCancel
        worker.tr.stop = message('nnet:trainingStop:UserCancel');
        calcNet = worker.originalNet;
      elseif (worker.perf <= worker.param.goal)
        worker.tr.stop = message('nnet:trainingStop:PerformanceGoalMet');
        calcNet = worker.best.net;
      elseif (worker.epoch == worker.param.epochs)
        worker.tr.stop = message('nnet:trainingStop:MaximumEpochReached');
        calcNet = worker.best.net;
      elseif (current_time >= worker.param.time)
        worker.tr.stop = message('nnet:trainingStop:MaximumTimeElapsed');
        calcNet = worker.best.net;
      elseif (worker.gradient <= worker.param.min_grad)
        worker.tr.stop = message('nnet:trainingStop:MinimumGradientReached');
        calcNet = worker.best.net;
      elseif (worker.mu >= worker.param.mu_max)
        worker.tr.stop = message('nnet:trainingStop:MaximumMuReached');
        calcNet = worker.best.net;
      elseif (worker.val_fail >= worker.param.max_fail)
        worker.tr.stop = message('nnet:trainingStop:ValidationStop');
        calcNet = worker.best.net;
      end
      % Training Record
      worker.tr = nnet.trainingRecord.update(worker.tr, ...
        [worker.epoch current_time worker.perf worker.vperf worker.tperf worker.mu worker.gradient
worker.val_fail]);
      worker.statusValues = ...
        [worker.epoch,current_time,worker.best.perf,worker.gradient,worker.mu,worker.val_fail];
      end
```

```
function [worker,calcNet] = trainingIteration(worker,calcLib,calcNet)
% Cross worker control variables
muBreak = [];
perfBreak = [];
```