THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

# TIME SERIES ANALYSIS FOR DYNAMIC RESOURCE PROVISIONING IN CLOUD PLATFORMS

**Master's Thesis**

**FATIH KÜÇÜKKARA**

**İSTANBUL, 2018**

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

INSTITUTE OF SCIENCE

COMPUTER ENGINEERING

# TIME SERIES ANALYSIS FOR DYNAMIC RESOURCE PROVISIONING IN CLOUD PLATFORMS

**Master's Thesis**

**FATİH KÜÇÜKKARA**

**Thesis Supervisor: ASSIST. PROF. TEVFİK AYTEKİN**

**İSTANBUL, 2018**

**T. C.**

**BAHÇEŞEHİR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**
**COMPUTER ENGINEERING**

Title of Thesis: Time Series Analysis for Dynamic Resource Provisioning in Cloud Platforms

Name and Surname of Student: Fatih KÜÇÜKKARA
Date of Thesis Defence: 23.05.2018

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assist. Prof. Yücel Batu SALMAN
Manager of Institute
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Tarkan AYDIN
Program Coordinator
Signature

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Member Signature

Thesis Supervisor
Assist. Prof. Tevfik AYTEKİN …………………….

Member
Assoc. Prof. Burcu TUNGA
…………………….

Member
Assoc. Prof. Mehmet Alper TUNGA
…………………….

## ACKNOWLEDGEMENTS

I would like to thank you to my supervisor for his great advises and to my company for their support and to my wife for her patience to my long work hours.

İstanbul, 2018 Fatih KÜÇÜKKARA

# ABSTRACT

## TIME SERIES ANALYSIS FOR DYNAMIC RESOURCE PROVISIONING IN CLOUD PLATFORMS

Fatih Küçükkara

Computer Engineering

Thesis Supervisor: Assist. Prof. Tevfik Aytekin

May 2018, 56 pages

The migration from local servers to the Cloud Data Centers has accelerated with the increasing trust of the companies to the Cloud Computing. Thanks to the virtualization technology, the service providers have started to give the Dynamic Resource Provisioning service for the fluctuating customer resource demands. At this context, keeping the resource balance between the peak hours and the off-peak hours in an efficient manner is a challenging issue for the service providers.

To handle this problem, a large number of Data Centers (DC) manage their customers' resources still in a static way. A number DCs however, use horizontal scaling. Few DCs, on the other hand, use vertical or hybrid scaling.

Nearly all of the methodologies mentioned above have certain drawbacks. Static management based on the peak times leads to the resource wastage at other times. Horizontal scaling may result in also wastage on some of the resource types customer may not need. Although vertical scaling is better, resizing without reboot and the resources of the host machine are important points in that way. Hybrid scaling is the best way because it uses both vertical and horizontal scaling in accordance with requirements, however, without prediction, even this method is not efficient enough due to the scaling overheads such as lack of knowledge about scaling amount.

The goal of this paper is to provide predictive hybrid scaling system on cloud. Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) are used as forecasting methods on the VMware and OnApp cloud platforms. Although the system is eligible for nearly all of the resource types, memory is selected for the experiments which have promising results.

**Keywords:** Dynamic Resource Provisioning, Hybrid Scaling, LSTM, ARIMA

# ÖZET

## BULUT TABANLI ORTAMLARDA DİNAMİK KAYNAK SAĞLAMA İÇİN ZAMAN SERİSİ ANALİZİ

Fatih Küçükkara

Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Tevfik Aytekin

Mayıs 2018, 56 sayfa

Yerel sunuculardan Bulut Veri Merkezlerine geçiş, şirketlerin Bulut Bilişim'e karşı artan güvenleriyle hızlandı. Sanallaştırma teknolojisi sayesinde servis sağlayıcılar, dalgalanan müşteri kaynak talepleri için Dinamik Kaynak Sağlama servisini vermeye başladılar. Bu bağlamda, yoğun saatlerle yoğun olmayan saatler arasındaki dengeyi verimli bir şekilde kurmak servis sağlayıcılar oldukça zorlayıcı bir iştir.

Bu durumla baş edebilmek için çok sayıda veri merkezi müşteri kaynaklarını statik bir şekilde yönetir. Ancak, bazı veri merkezleri yatay ölçekleme kullanırken bazıları da dikey ya da hibrit ölçekleme kullanmaktadırlar.

Yukarıda bahsedilen yöntemlerden hemen hemen hepsinin bazı dezavantajlara sahiptir. Yoğun zamana göre ayarlanmış statik yönetim, diğer zamanlar için kaynak israfı oluştutur. Yatay ölçekleme, müşterinin ihtiyacı olmayan bazı kaynak tipleri için kaynak israfı oluşturabilir. Dikey ölçekleme daha iyi bir yol olmasına rağmen yeniden başlatma olmadan kaynak değişimi ve ana makine kaynakları bu yöntemde dikkat edilmesi gereken noktalardır. İhtiyaca göre yatay ve dikey ölçekleme yöntemlerini kullanabildiği için hibrit ölçekleme en iyi yoldur ancak tahminleme olmadan arttırılacak kaynak miktraının bilinememesi gibi bazı ek yükler sebebiyle bu yöntem bile yeterince verimli değildir.

Bu çalışmanın amacı, bulut üzerinde tahminleme yapabilen hibrid ölçeklendirme sistemi kurmaktır. Uzun Kısa Süreli Bellek ve Otoregresif Entegre Hareketli Ortalama tahminleme yöntemleri, VMware ve OnApp bulut platformları için uyarlanmıştır. Sistem neredeyse tüm kaynak tipleri için çalışabilecek yetkinliğe sahip olsa da testler için bellek seçilmiştir. Çalışmalar başarılı sonuçlar üretmiştir.

**Anahtar Kelimeler:** Dinamik Kaynak Sağlama, Hibrit Ölçekleme

## CONTENTS

# TABLES

## FIGURES

# ABBREVIATIONS

| | | |
|---|---|---|
| API | : | Application Programming Interface |
| AR | : | Autoregression |
| ARIMA | : | Autoregressive Integrated Moving Average |
| BNN | : | Backpropagation Neural Network |
| BPAAS | : | Business Process AS A Service |
| CAGMA | : | Critical Amount Guaranteed Memory Allocation |
| CDN | : | Content Delivery Network |
| CMA | : | Critical Memory Amount |
| DC | : | Data Center |
| DDHPA | : | Data-Driven Hybrid Prediction Approach |
| DRAAS | : | Disaster recovery as a service |
| EWMA | : | Exponentially Weighted Moving Average |
| HW | : | Hardware |
| I | : | Integrated |
| IAAS | : | Infrastructure As A Service |
| ICC | : | Information Collector Component |
| IPMI | : | Intelligent Platform Management |
| LSTM | : | Long Short Term Memory |
| MA | : | Moving Average |
| MAC | : | Memory Allocator Component |
| MAPE | : | Mean Absolute Percentage Error |
| MMC | : | Memory Monitoring Component |
| MSD | : | Memory Statistics Database |
| NN | : | Neural Network |
| OS | : | Operating System |
| OM | : | Orchestration Module |
| PAAS | : | Platform As A Service |
| PM | : | Physical Machine |
| PSO | : | Particle Swarm Optimization |

| | | |
|---|---|---|
| RVLBPNN | : | Rand Variable Learning Rate Backpropagation Neural Network |
| RNN | : | Recurrent Neural Network |
| RMSE | : | Root Mean Square Error |
| SAAS | : | Service As A Service |
| SAN | : | Storage Area Network |
| SLA | : | Service Level Agreement |
| SLO | : | Service Level Objective |
| SNMP | : | Simple Network Management Protocol |
| VM | : | Virtual Machine |
| VMM | : | Virtual Machine Monitor |
| QOS | : | Quality Of Service |

# SYMBOLS

| | | |
|---|---|---|
| Input of Recurrent Neural Network | : | $x_t$ |
| Weight of Input of Recurrent Neural Network | : | $W$ |
| Output of Recurrent Neural Network | : | $h_t$ |
| Previous Output of Recurrent Neural Network | : | $h_{t-1}$ |
| Weight of Previous Output of Recurrent Neural Network | : | $U$ |
| Activation Function | : | $\sigma$ |
| Old State | : | $f_t$ |
| Forget Bias, Input Bias, Output Bias | : | $b_f$, $b_i$, $b_o$ |
| Input State | : | $i_t$ |
| Forget Weight, Input Weight, Output Weight | : | $W_f$, $W_i$ |
| Cell State | : | $C_t$ |
| Previous Cell State | : | $C_{t-1}$ |
| Multiplier for scaling 0-1 | : | $tanh$ |
| Output State | : | $O_t$ |
| Noise | : | $\omega_t$ |
| Time | : | $(t)$ |
| Estimated Load | : | $E(t)$ |
| Observed Load | : | $O(t)$ |
| Trade-off between Stability and Responsiveness | : | $\alpha$ |

# 1. INTRODUCTION

Cloud Computing is the shining technology of this era. With the advance of elasticity, the lack of massive initial capital and no need to know about the complexity of the service infrastructure, companies have begun to think about to transfer their dedicated resources to Cloud Data Centers. (Xiao, et al., 2013). This transformation includes hardware shift - Infrastructure as a Service (IaaS) such as virtual server, business platform shift - Platform as a Service (PaaS) such as Microsoft Dynamics CRM and software application shift – Software as a Service (SaaS) such as email service (Matthias, et al., 2016).

Since this is an appealing market that is seen from *Table 1.1*, the count of the service providers with a hot competition has increased gradually at last decades, Microsoft, Amazon and Google are just the several of the global ones. In order to survive, service providers struggle with using all features of the Cloud Base Environments and working hard to enhance the services of this platform.

**Table 1.1: Worldwide public cloud services revenue forecast (billions of USD)**

| | **2016** | **2017** | **2018** | **2019** | **2020** |
|---|---|---|---|---|---|
| **Cloud Business Process Services (BPaaS)** | 39.6 | 42.2 | 45.8 | 49.5 | 53.6 |
| **Cloud Application Infrastructure Services (PaaS)** | 9.0 | 11.4 | 14.2 | 17.3 | 20.8 |
| **Cloud Application Services (SaaS)** | 48.2 | 58.6 | 71.2 | 84.8 | 99.7 |
| **Cloud Management and Security Services** | 7.1 | 8.7 | 10.3 | 12.0 | 13.9 |
| **Cloud System Infrastructure Services (IaaS)** | 25.4 | 34.7 | 45.8 | 58.4 | 72.4 |
| **Cloud Advertising** | 90.3 | 104.5 | 118.5 | 133.6 | 151.1 |
| **Total Market** | 219.6 | 260.2 | 305.5 | 355.6 | 411.4 |

*Source*: (Stamford, 2017)

On-demand subscription is a very powerful feature of Cloud-Based Services helping companies change their continual business models (Lu, et al., 2016). One of the most known payment strategies is the Pay As You Go model that allows customers to customize their hardware or software resource needs and to provide payment options according to their usage. (Techopedia, 2018). According to the (Zhang, et al., 2017), currently, there are two types of pricing models at the global market one of which is Pay As You Go that is mentioned above and the other of which is the reserved subscription model that the customer gives an initial fee and pays monthly by a discounted specialized price and Amazon EC2 is a sample provider for these models. Although companies choose the proper model based on their budget and business types, there is a possible problem that they may encounter which is resource customization according to the varying workload at different times.

This is a competitive market and providers try to do their best for serving the best quality of service to their customers in terms of pricing and the service itself for the cases of fluctuating resource demands, attackers, hardware failures and so on. Keeping the balance with minimum pricing and the best possible service quality is a very challenging task for service providers. To cope with the resources that are alternating continually is a trending topic nowadays with the pay as you go model. Studies done on this subject show that many data centers waste their resources due to the static provisioning done based on the maximum resource demand to prevent shortage (Xiao, et al., 2013). When thinking about the times of minimum resource demand, a two-sided problem arises. From the service providers' point of view, this means an inefficient management of resource though they get enough money and this means an extra monetary cost from the customers' point of view though they guarantee the good level of service quality at peak times.

Auto-scaling is a very effective solution of these kinds of problems. According to the (Chenhao, et al., 2017), the definition of it is:

> *To efficiently utilize elasticity of clouds, it is vital to automatically and timely provision and de-provision cloud resources without human intervention, since overprovisioning leads to resource wastage and extra monetary cost, while under provisioning causes performance degradation and violation of service level agreement (SLA). This mechanism of dynamically acquiring or releasing resources to meet QoS requirements as called auto-scaling.*

On-demand usage and auto-scaling provide natural growing strategy for the companies. That is, when the customer portfolio of the company rises, the revenue of it rises and the expenses of it also rise automatically, which is the core of many success stories like Instagram (Vazquez, et al., 2015). Virtualization, enabling hardware resources to be used as a number of Virtual Machines (VM) which may have different Operating Systems (OS) is the heart of auto-scaling. As (Matthias, et al., 2016) state that to provide better Quality of Service (QoS) and to use hardware resources more efficiently, using VMs is an effective way. Moreover, Green Computing namely energy-efficient computing can be reached with the help of advanced features of virtualization such as hot resize meaning that changing the resource of VM without rebooting and live migration meaning that transfer VMs to other hosts resulting in closure of the underutilized ones (Lu, et al., 2016).

Depending on the cloud platforms including host machines, hypervisors and operating systems of VMs, there are three ways of auto-scaling, one of which is the Horizontal Scaling – creating or destroying VM instances, the other of which is the Vertical Scaling – VM resource resizing and the last of which is the Hybrid Scaling which is the mix usage of the first two ways (Lu, et al., 2016). All of them that have some advantages and disadvantages can be used based on the requirements. Horizontal Scaling, very common among the service providers and probably the most used type of scaling, is briefly increasing or decreasing the instance count according to the demand rate. It is easy to use however it may be ineffective if the resource demand is small or partial. That is to say, when we create an instance it will be a bundle of CPU, ram, disk and etc. If our system is only in need of CPU for example, the other resources will be used inefficiently. Vertical scaling, however, is not so common because changing the resources like memory or CPU without reboot is a problematic and challenging issue depending on the hypervisors and the OSs of VMs. The resource may be changed after VM is shutting down but this may lead to the violation of the SLA. However, it will be more effective and energy saving approach than the horizontal one if the system does not fluctuate in an extraordinary way. That is to say, vertical scaling is limited with the resources of the Physical Machine (PM). If the demand exceeds the edges of PM, the VM should be transferred to other PM. The Hybrid Scaling is the mix of the horizontal and vertical ones, which benefits the pros of them avoiding the cons. The idea is to use the vertical scaling to the edges of the host machine and to change into the horizontal one when it is needed, which is the most effective way. However, there are other vital problems about scaling without the type of it which are the amount of resource to be resized, the time of change and the duration of the process not to affect the service quality and not to violate the SLA.

To cope with these problems, the system should forecast the correct amount of the VM by using historical data with some predictive algorithms and provision or de-provision the related resource before the real need comes out without the reboot (Vazquez, et al., 2015).

Accurate prediction is a vital step at this point Long Short-Term Memory (LSTM) Neural Networks (NN) and Autoregressive Integrated Moving Average (ARIMA) are the two of best-known algorithms which are based on the Time Series Analysis for predicting future resource needs. Studies showed that LSTM is excelling thanks to the prediction abilities (Toque, et al., 2016).

The aim of this paper is to make the best predictions with the help of LSTM and ARIMA, to resize a VM resource without reboot and to create new VM instances based on the hybrid scaling. Although the system can be used with nearly all of the VM resources, memory is selected for the simulations. VMware and OnApp are used as selected cloud platforms.

**The key contributions** of this paper are:

i. Development of a prediction module using LSTM
ii. Development of a prediction module using ARIMA
iii. Evaluation the performances of LSTM  and ARIMA
iv. Development of the Dynamic Provision System on VMware Platform
v. Development of the Dynamic Provision System on OnApp Platform
vi. Evaluation of the OnApp and VMware Cloud Platforms regarding both horizontal and vertical scaling, namely hybrid scaling from different perspectives such as duration and hot resize.

The rest of the paper is organized as follows: Cloud Computing, Virtualization, VMware, OnApp, Zabbix, Service Level Agreement, Time Series Prediction, LSTM and ARIMA are given as background information so as to explain the research in the best manner in Section 2, Related Works from literature are given in Section 3, Data and the Dynamic Resource Provisioning are explained in Section 4, Results and Discussion are stated in Section 5 and Conclusion is placed in Section 6 lastly.

# 2. BACKGROUND

## 2.1 CLOUD COMPUTING

Cloud Computing is a computing paradigm in which a number of internetworked virtualized servers that are parallel or distributed, scalable and unified are used to allow sharing computer resources, storing data and serving some services through the internet to the public, private or hybrid customer groups with SLAs (Gupta, et al., 2017) (İsmail & Riasetiawan, 2016) (Springer, 2010).

Cloud Computing prevents companies to make upfront research and development about computer systems as well as massive investments about it before starting up a business. The goal of this environment is to make companies focus only on their core businesses rather than setting up complex computer infrastructure, employing a team to maintain it and cost of them (Matthias, et al., 2016). With the help of SLAs, companies guarantee the level of service quality they require from the response time of a website to situation of a system in the most complicated disaster scenarios. When we think about the pricing issues, major is Pay As You Go model where companies pay whatever they use. That is, when they use a small amount of resources they pay a small amount of fee, when they increase the resources with their increasing customer portfolio, their fee is automatically increasing, which is natural growth and efficient extending policy.

### 2.1.1 Characteristics of Cloud Computing

There are several characteristic features of the Cloud Computing that serves a number of benefits to the companies in terms of elasticity, access, speed, cost and transparency.

#### 2.1.1.1 On-Demand Self-Service

It refers on-demand elasticity of the system with the request of the customer without human interaction (Mell & Grance, 2009) (Rouse, 2017).

#### 2.1.1.2 Broad Network Access

Services are provided to consumers through the internet which enables clients to use them through their private computers, VMs, tablets and so on (Mell & Grance, 2009).

#### 2.1.1.3 Resource Pooling

Provisioned resources including both hardware and software are pooled that is consumers mostly do not know the exact locations of the PMs of their VMs. They may know only the country or the city. This is about the multitenant face of the system (Mell & Grance, 2009).

#### 2.1.1.4 Rapid Elasticity

This feature provides automatic or manual quick provision or de-provision to prevent any kind of performance problem which provides the cost efficiency to the companies (Mell & Grance, 2009) (Ranger, 2018).

#### 2.1.1.5 Measured Service

Without the type of service, usage is monitored and reported, which is important since the pay as you go is the major pricing model and customers should know what they use to pay (Mell & Grance, 2009).

### 2.1.2 Service Models of Cloud Computing

Cloud Computing Services are mainly categorized under three main headings which are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

#### 2.1.2.1 Software as a Service (SaaS)

It means that some web services which are running and maintained in the cloud by technical guys are accessed through the internet and consumers know nearly nothing about the infrastructure of it and has no control to manage it. Generally, subscription-based pricing is served and email and dictionary are two samples of it (Mell & Grance, 2009) (Rouse, 2017).

#### 2.1.2.2 Platform as a Service (PaaS)

It mainly targets the software developers providing a platform through the web portal, Application Programming Interface (API) and so on for the developers to create their own applications with using some kinds of programming languages or tools supported by the service provider. The consumer has no control over the infrastructure like network or storage of the service and Microsoft's Azure and Salesforce's Force.com are some examples of this service (Mell & Grance, 2009) (Rouse, 2017).

#### 2.1.2.3 Infrastructure as a Service (IaaS)

It provides lots of power to the consumers from VM provisioning, to the network and firewall settings. Including Operating System settings, consumers have control nearly everything except the cloud infrastructure. Service providers mostly serve some tools such as web APIs or web portals for consumers to use. VMs provided by the cloud data centers are the major example of this type of service (Mell & Grance, 2009) (Rouse, 2017).

### 2.1.3 Deployment Models of Cloud Computing

Cloud deployment models mainly categorized under four headings which are private, community, public and hybrid (Mell & Grance, 2009).

#### 2.1.3.1 Private Cloud

It means that the cloud system is used exclusively by a company on a private network. In that type, the system resources can be maintained in the consumers' own data centers (Mell & Grance, 2009) (Azure, 2018)

#### 2.1.3.2 Community Cloud

It serves a system for some organizations that is a part of a community having same missions, worries and interests. The resources can be both on-premise or off-premise (Mell & Grance, 2009).

#### 2.1.3.3 Public Cloud

It serves a system through the internet to access of everybody which sold on-demand and the pricing policy can be per minute, hour and so on (Mell & Grance, 2009) (Rouse, 2017) (Springer, 2010).

#### 2.1.3.4 Hybrid Cloud

It is a mix of cloud types mentioned above that are private, community and public. Companies use a private cloud for their special requirements and use other types based their needs (Mell & Grance, 2009) (Rouse, 2017).

## 2.2 VIRTUALIZATION

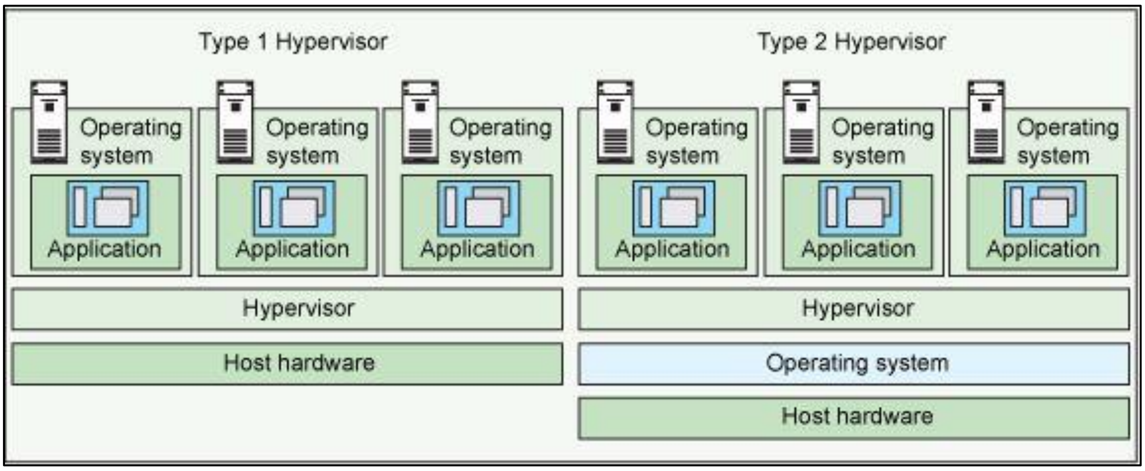
Virtualization is the key component of Cloud Computing enabling a PM acts as multiple isolated VMs with different OSs that can be easily provisioned or de-provisioned by multiplexing the underlying hardware such as CPU and memory thanks to the hypervisors (Gupta, et al., 2017) (Wu & Sun, 2017). Using virtualization provides consolidation of resources, reduction in the hardware cost, elasticity, speed and so on (Tholeti, 2011).

Hypervisor named also as Virtual Machine Monitor (VMM) is a kind of software responsible for creation and management of VMs. Hypervisors are the heart of the virtualization, form hardware multiplexing to task management nearly every critical point about virtualization is done here.

Hypervisors are mainly categorized under two headings which are Type 1 (Bare Metal) and Type 2 (Hosted) that described in *Figure 2.1*. Bare Metal ones such as VMware ESXi and Xen are running directly on host's hardware (HW) to control HW and guest's OSs. Hosted ones however such as VMware Workstation and Virtual box running on an OS on the HW and separates the guest OSs from it (Tholeti, 2011).

Hypervisors maps VMs and PMs and consumers do not know that where their VM instances run exactly. The may only know about the country or the city (Xiao, et al., 2013). This situation is the result of the pooling structure of the cloud computing.

**Figure 2.1: Types of hypervisors**



*Source:* (Tholeti, 2011)

### 2.2.1 Power VM

Power VM known as Advanced Power Virtualization is a limitless server virtualization which is a feature of IBM POWER5, POWER6, POWER7 and POWER8 servers provides secure and elastic virtualization environment for AIX, IBM I and Linux applications (Tholeti, 2011) (IBM, 2018)

### 2.2.2 VMware ESXi Server

VMware ESXi is a bare metal hypervisor which accesses and manages hardware resources directly, increasing consolidation ratios and making it better from the hosted ones (VMware, 2018).

### 2.2.3 Xen

Xen is an open source bare-metal hypervisor for IA-32, x86-64, Itanium and ARM architectures in order to run many instances of an OS which increases server utilization, consolidate server farms and so on (XenProject, 2018)

### 2.2.4 Kvm

Kernel-based Virtual Machine is a kind of virtualization system for the Linux kernel. It supports native virtualization on processors that have hardware virtualization extensions and can be used with lots of guest OS including variations of Linux, Solaris, Windows Haiku and so on (Tholeti, 2011) (Kvm, 2018)

### 2.2.5 z/VM

z/VM hypervisor is designed for IBM’s Z and Linux One servers supporting a wide variety of OSs including ZOs and Linux to enable organizations to run thousands of Linux servers on single IBM Z Systems or Linux ONE server with the highest efficiency (IBM, 2018) (IBM, 2018).

## 2.3 VMWARE

VMware is a branch of Dell Technologies to setup in order to provide the Virtualization solutions. The company has both hosted and bare metal hypervisors.

### 2.3.1 VMware Desktop Applications

There are different kind of applications for Windows, Linux and Mac OSs.

#### 2.3.1.1 VMware Workstation

VMware Workstation is a desktop application which is a Hosted Hypervisor to provide multiple VM instances on Windows and Linux OSs (techopedia, 2018).

#### 2.3.1.2 VMware Fusion

VMware Fusion is a desktop application which has the same purpose with VMware Workstation but for Mac users (techopedia, 2018).

#### 2.3.1.3 VMware Player

VMware Player is a desktop application which has the same purpose with VMware Workstation but the free version of it (techopedia, 2018).

### 2.3.2 VMware Server Applications

There are also Type 1 and Type 2 Server type of hypervisors on VMware.

#### 2.3.2.1 VMware ESX Server

VMware ESX Server is a Bare Metal Hypervisor which is an Enterprise Solution (techopedia, 2018).

#### 2.3.2.2 VMware ESXi Server

VMware ESXi Server is the same with the VMware ESX Server, however, its service console is replaced with Busy Box installation and needs low disk space to run (techopedia, 2018).

#### 2.3.2.3 VMware Server

VMware Server is a freeware Type 2 – Hosted Hypervisor that can be used different OSs (techopedia, 2018).

There are another VMware management products, vSphere is one of the most important ones which is used in our simulation. VSphere is a suite consisting of products below: (VMware, 2018):

**ESXi:** It is a Type 1 Hypervisor mentioned in 2.3.2.2.

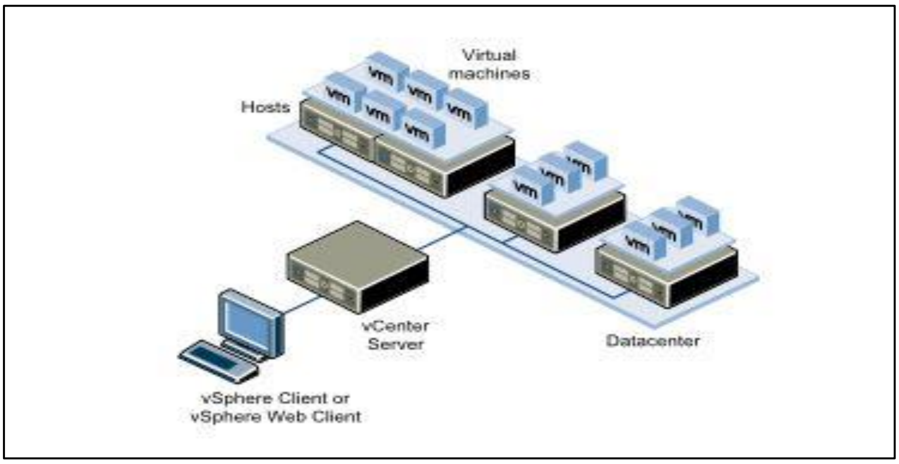
**Vsphere Web Client:** It is a web browser interface to inform administrative tasks including management of VMs, console connections and so on.

**Vcenter Server:** It unifies the hosts' resources to be used in VMs.

**Host:** It is a PM having one of the hypervisors mentioned in 2.3.2.

**Virtual Machine:** It is a virtual computer running OS and software applications.

**Figure 2.2: The vsphere suit**

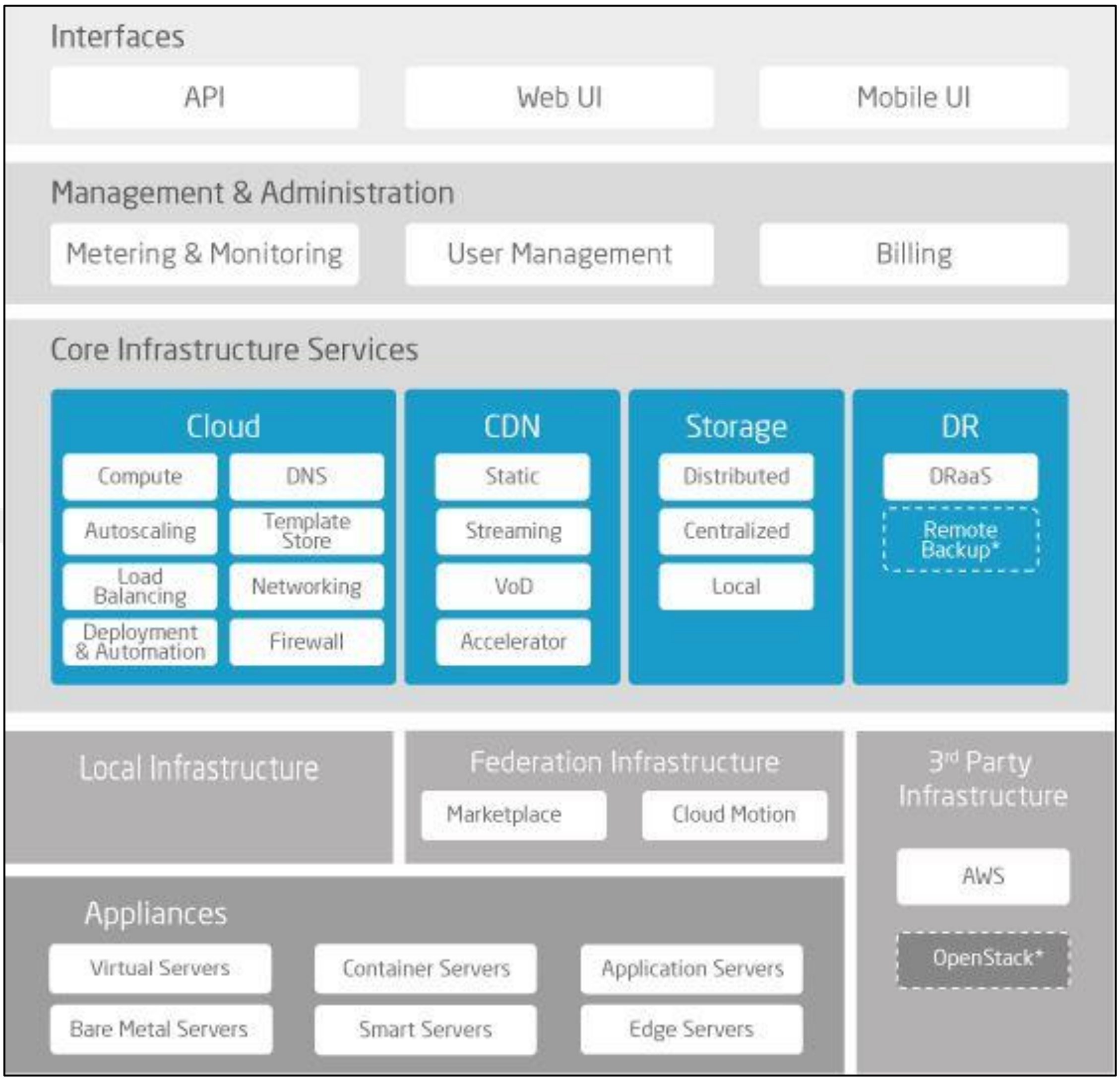


*Source:* (VMware, 2018)

## 2.4 ONAPP

OnApp is a Cloud Management Solution for service providers specifically for Data centers and enterprises managing their own cloud systems. It enables companies to serve IaaS, Content Delivery Network (CDN), storage, disaster recovery and so on. It is compatible with Xen, KVM and VMWare hypervisors (Butler, 2013) (Hostbill, 2018).

**Figure 2.3: The OnApp architecture**



*Source:* (OnApp, 2018)

### 2.4.1 OnApp Cloud

OnApp is one of the leading public cloud platforms serving to variety of users with millions of VMs. The System which supports VMware, Xen, KVM and EC2, provides highly scalable, available, customizable management portal gives service on public, private and hybrid cloud environments (OnApp, 2018)

### 2.4.2 OnApp for VMware Clouds

OnApp serves some of its features which are functionality, easy of use, easy of management, automation, billing and so on to some VMware management products, vCenter and vCloud Director (OnApp, 2018).

### 2.4.3 OnApp Storage

It serves high capability distributed Storage Area Network (SAN) that built into OnApp Platform in order provide easy, fast and scalable storage for the cloud (OnApp, 2018).

### 2.4.4 OnApp Cdn

OnApp Content Delivery Network (CDN) service including CDN software stack and custom Any-cast DNS service provides public or private CDNs for service providers or enterprises (OnApp, 2018).

### 2.4.5 OnApp DraaS

O OnApp napp Disaster Recovery as a Service (DRaaS) provides an affordable and high-performance disaster recovery solution to companies (OnApp, 2018).

### 2.4.6 OnApp App Server

It is kind of add-on library giving access to users to reach the SaaS and PaaS kind of applications (OnApp, 2018).

### 2.4.7 OnApp Containers

OnApp provides a framework for the use of Docker in the clouds. Container Servers provide the management facilities and tools for the clients to run Docker and Kubernetes of a Standard OnApp VM (OnApp, 2018).

### 2.4.8 OnApp Accelerator

It is an add-on providing several performance improving features such as page, script, image optimizations and publishing these resources to global CDN includes 19 locations from OnApp Federation (OnApp, 2018).

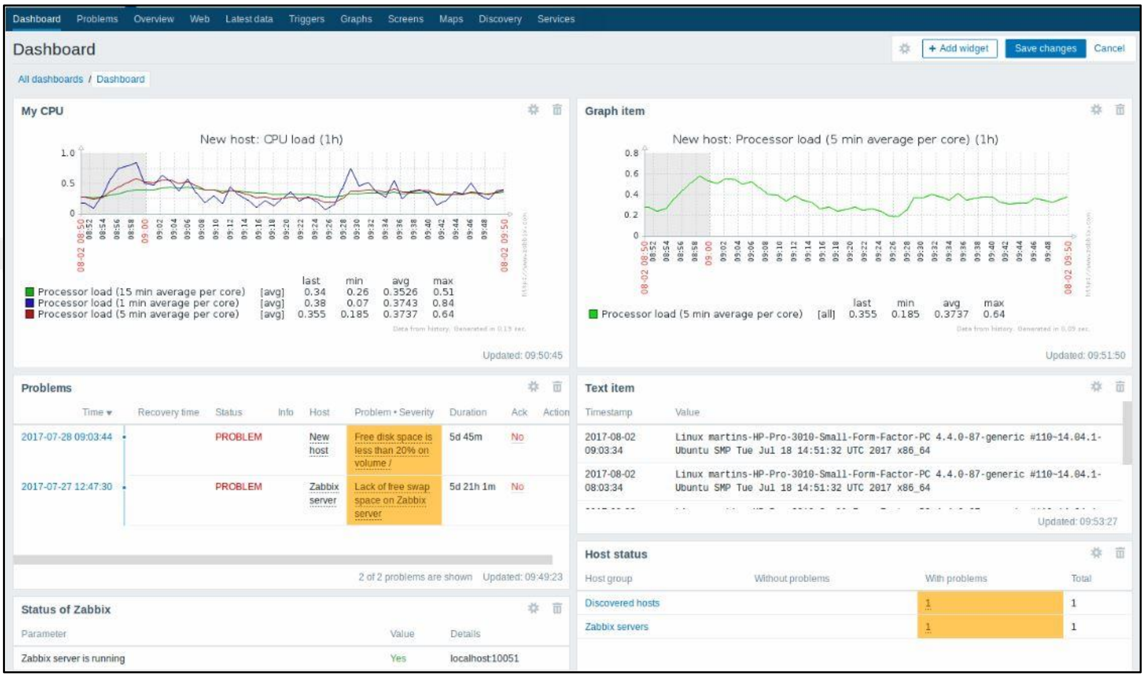
## 2.4 ZABBIX

Zabbix is cross-platform open source monitoring system that is capable of monitoring nearly everything such as servers, network and etc. It stores and visualized the monitored data. It also enables customers to set thresholds and get notifications about them.

Linux based Zabbix server collects monitoring data such as CPU load, network utilization, free available memory and so on from its agents. If we do not use the agent, the system supports certain generic protocols which are Simple Network Management Protocol (SNMP), Intelligent Platform Management Interface (IPMI) (Rouse, 2013).

In this research, Zabbix is used to get historical resource data for the calculation of the future resource forecasting. Sample ram usage view can be seen in *Figure 2.4*.

**Figure 2.4: Zabbix sample view**



| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 Processor load (15 min average per core) | [avg] | 0.34 | 0.26 | 0.3526 | 0.51 |
| 🟦 Processor load (1 min average per core) | [avg] | 0.38 | 0.07 | 0.3743 | 0.84 |
| 🟥 Processor load (5 min average per core) | [avg] | 0.355 | 0.185 | 0.3737 | 0.64 |

Updated: 09:50:45

| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 Processor load (5 min average per core) | [all] | 0.355 | 0.185 | 0.3737 | 0.64 |

Updated: 09:51:50

Problems

| Time ▾ | Recovery time | Status | Info | Host | Problem • Severity | Duration | Ack | Action |
|---|---|---|---|---|---|---|---|---|
| 2017-07-28 09:03:44 | | PROBLEM | | New host | Free disk space is less than 20% on volume / | 5d 45m | No | |
| 2017-07-27 12:47:30 | | PROBLEM | | Zabbix server | Lack of free swap space on Zabbix server | 5d 21h 1m | No | |

2 of 2 problems are shown Updated: 09:49:23

Text item

| Timestamp | Value |
|---|---|
| 2017-08-02 09:03:34 | Linux martins-HP-Pro-3010-Small-Form-Factor-PC 4.4.0-87-generic #110-14.04.1-Ubuntu SMP Tue Jul 18 14:51:32 UTC 2017 x86_64 |
| 2017-08-02 08:03:34 | Linux martins-HP-Pro-3010-Small-Form-Factor-PC 4.4.0-87-generic #110-14.04.1-Ubuntu SMP Tue Jul 18 14:51:32 UTC 2017 x86_64 |

Updated: 09:53:27

Status of Zabbix

| Parameter | Value | Details |
|---|---|---|
| Zabbix server is running | Yes | localhost:10051 |

Host status

| Host group | Without problems | With problems | Total |
|---|---|---|---|
| Discovered hosts | | 1 | 1 |
| Zabbix servers | | 1 | 1 |

*Source*: (Zabbix, 2017)

## 2.5 SERVICE LEVEL AGREEMENT

Service Level Agreement can be defined as the boundaries of a service quality that a customer is waiting from a supplier. This agreement should be based on certain measurable and verifiable metrics to be able to make an assessment and include compensation items for the sake of mainly customers. Rise of the usage of cloud services make providers to increase SLA standards to appeal more customers (Vazquez, et al., 2015).

According to the Standardization, SLAs have Service Level Objectives (SLO) at different categories and sub-categories which are Performance including Availability, Response Time, Capacity, Capability Indicators, Support and so on, Security including Reliability, Authentication - Authorization, Cryptography and so on, Personal Data

Protection including Purpose Specification, Data Minimization, Accountability and so on (DigitalSingleMarket, 2014).

## 2.6 TIME SERIES PREDICTION

Time Series means any kind of observation data depending on time which is usually ordered. It can be used in the various fields such as meteorology for temperature prediction, marketing for sales prediction or stock forecasting and so on (Bontempi, 2013).

Time Series Prediction, however, is a probabilistic and statistical approach taking historical data into account to produce a model in order to forecast a future value with the help of certain algorithms such as LSTM. *Table 2.1* shows us a time series data sample getting from a VM.

**Table 2.1: Time series data – free memory**

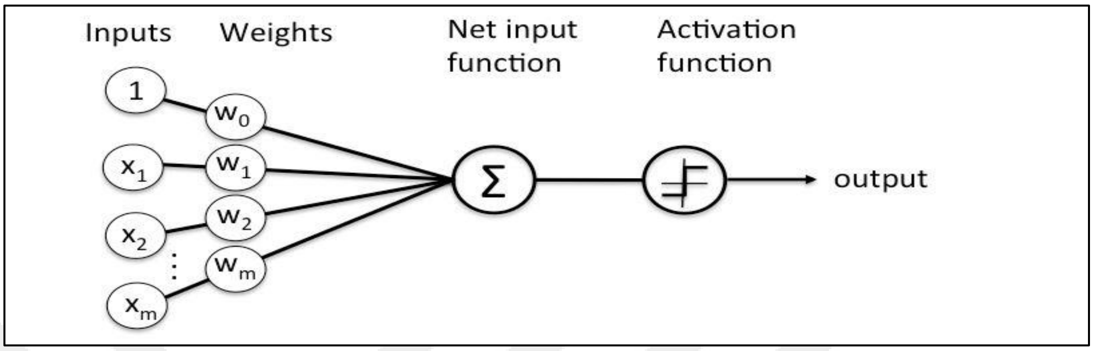
| **Date** | **Time** | **Timestamp** | **Value (Byte)** |
|---|---|---|---|
| **2018-04-16** | 20:09:53 | 1523898593 | 11974021120 |
| **2018-04-16** | 20:08:01 | 1523898481 | 11821961216 |
| **2018-04-16** | 20:06:47 | 1523898407 | 12012388352 |
| **2018-04-16** | 20:05:56 | 1523898356 | 11616415744 |
| **2018-04-16** | 20:03:57 | 1523898237 | 11515367424 |
| **2018-04-16** | 20:01:57 | 1523898117 | 11764895744 |

## 2.7 LONG SHORT TERM MEMORY

Long Short-Term Memory (LSTM) developed by Hochreiter and Schmidhuber in 1997 is a special variant of Recurrent Neural Networks (RNN) which is a complex type of Neural Networks (NN). Let us explain from bottom to up to describe more clearly.

NNs are artificial learning models that are inspired by the structure of the human biological nervous system (Stergiou & Siganos, 1989). NNs are composed of the layers which are input, hidden and output each of which has nodes working as a computation center and creates outputs for the next layer (DL4J, 2017). As illustrated in *Figure 2.5,* nodes collect inputs multiplied with weights and produce outputs according to the activation function. In that structure input resembles the example that is used to train the system, hidden layer means the complex area that is busy with computational works and the output layer is the product of the system (Thomas, 2017).

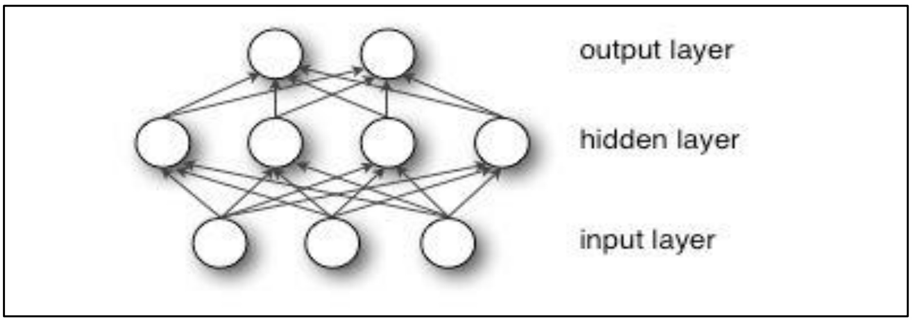
**Figure 2.5: The structure of a node**



*Source:* (DL4J, 2017)

The structure of the NN layers is described in *Figure 2.6*. Each layer transfers the outputs to the next layer after the required computations.

**Figure 2.6: The structure of the layers**



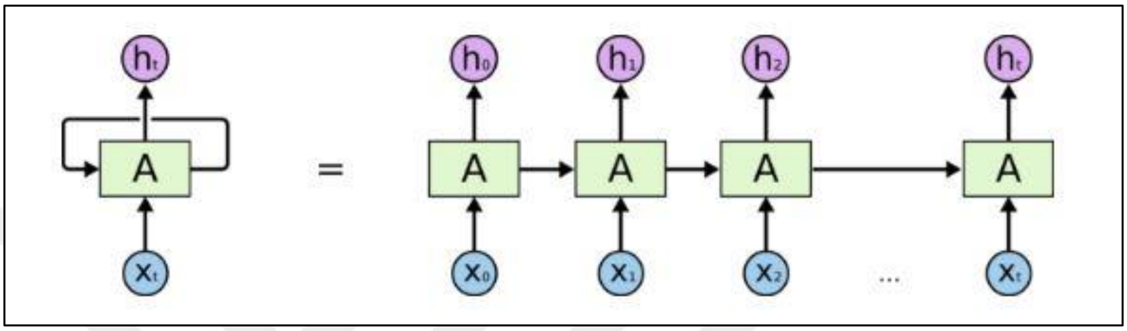
*Source:* (DL4J, 2017)

RNNs are more complicated and extended version of NNs, which can use the outputs of the nodes as inputs again. After a while this situation resembles that the RNN seems to have memory (DL4J, 2017), which can be stated mathematically as below:

$$h_t = \sigma(Wx_t + Uh_{t-1}) \tag{2.1}$$

Where $x_t$ is the present input and $W$ is the weight of it, $h_{t-1}$ is the previous output and $U$ is the weight of it, $h_t$ is the current output and $\sigma$ is the sum function. Since the input-output dependency RNN can handle sequence type of problems (Toque, et al., 2016). RNN can be successfully solved the language modelling kind of problems and illustrated as *Figure 2.7.*

**Figure 2.7: The view of RNN sequence**



*Source:* (Olah, 2015)

Although RNN is successful on complicated problems with short-term dependencies, it is not good enough with the long-term ones. That is why, LSTM comes out (Hochreiter & Schmidhuber, 1997). To keep the long term dependencies, LSTM introduces cell state as a manager deciding to remove, add or pass the information. This regulation is handled through the gates that are forgotten, input and output (Zhang, et al., 2018).

**Forget Gate:** This is the gate to decide which information is removed.

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

**(2.2)**

**Input Gate:** This is the gate to decide which information is stored.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$$

**(2.3)**

$$C_t = tanh(W_c.[h_{t-1}, x_t] + b_c)$$

**(2.4)**

$$C_t = f_t.C_{t-1} + i_t.C_t$$

**(2.5)**

**Output Gate:** This is the gate which information is sent as output.

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$

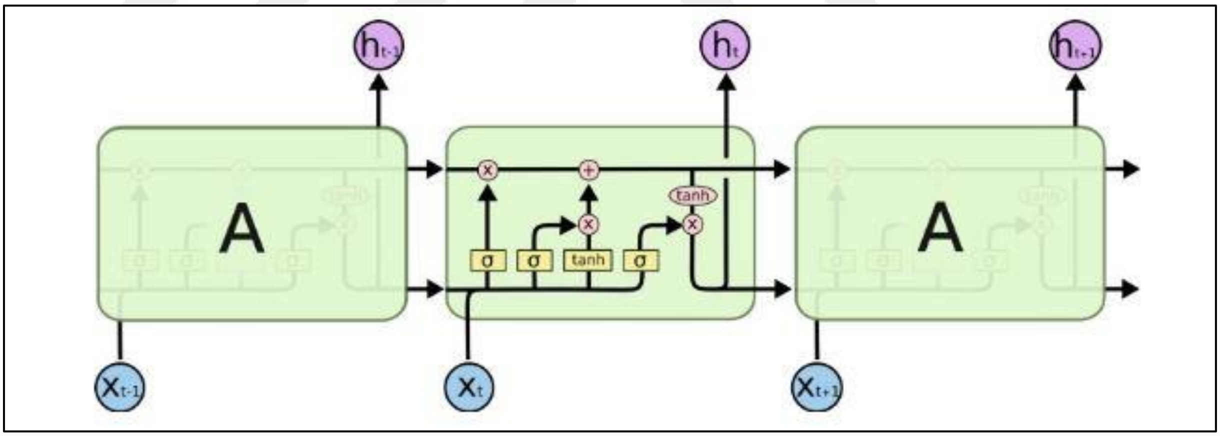
**(2.6)**

$$h_t = O_t.tanh.C_t$$

**(2.7)**

The overall structure of the LSTM is described in *Figure 2.8.*

**Figure 2.8: The structure of lstm**



*Source:* (Olah, 2015)

## 2.8 ARIMA

Autoregressive Integrated Moving Average (ARIMA) which is more complicated version of Autoregressive Moving Average (ARMA) is composed of three types of models that are Auto-regression (AR), Integrated (I) and Moving Average (MA) (Brownlee, 2017).

Auto-regression (AR(p)) is defined basically as the prediction of value from the previous ones from time series and the order of auto-regression means the lag value that is the preceding values to be used for the forecast (PSU, 2018). The formulation is described below:

$$X_t = \sum_{j=1}^{p} \sigma_j X_{t-j} + \omega_t \tag{2.8}$$

Where $\sigma$ is the model coefficients, p is a non-negative integer, t is time and $\omega_t$ is nose (Huerta, 2006).

Integrated (I(d)) means that using differences between observations and the past ones.

Moving Average (MA(q)) uses the relation between observations and errors coming from the past. It basically means that error of the model is the linear combination of past errors (Dalinina, 2013). The formulation is as the following:

$$X_t = \sum_{j=1}^{q} \emptyset_j \omega_{t-j} + \omega_t \tag{2.9}$$

Autoregressive Integrated Moving Average (ARIMA (p,d,q)) is the generalization of the methods mentioned above where p means the lag observations, d means the number of differences used and q means the size of moving average window (Brownlee, 2017).

# 3. RELATED WORK

Studies done on the Dynamic Resource Provisioning can be categorized under two headings which are reactive and proactive in other words predictive.

(Wu & Sun, 2017) proposed a reactive method for Virtualization Platforms which is called *Critical Amount Guaranteed Memory Allocation (CAGMA)* in order to manage memory dynamically. The main idea behind this method is to guarantee that the available memory of VMs is never less than *Critical Memory Amount (CMA)* that is updated periodically, at the time of swapping events. By doing this, the available memory will always be sufficient so the swapping events in which intense performance slowing I/O activities are done with disks can be reduced to considerable levels, which increases overall performance as well as efficiency. In that research, Xen platform which is the version 4.2.2 is used with a number of experimental studies and prosperous results are collected as a result.

(Shaikh & Shrawankar, 2015) studied on a global allocation technique on Virtualization for increasing or decreasing the memory of virtual machines on runtime, which is also reactive study because there is no any kind of prediction. In this research, Virtual Machine Monitor which is known also as hypervisor is divided into four components, the first one of which is Memory Monitoring Component (MMC) which is used to monitor the memory usage including both totals and used memories of the virtual machine. The second of which is Information Collector Component (ICC) which is used to get monitoring data from both Physical Machines and Virtual Machines. The third one of which is Memory Allocator Component (MAC) which is used to manage the virtual machines' memory amounts according to their usage level. That is, the system increases the amount of memory of the VM if it has memory shortage or the system get some of the memory of VM back if it has too much available memory. The fourth one of which is Memory Statistics Database (MSD) which is used to keep monitored data of physical machine and virtual machine, which can be used for a number statistical purposes. In that study, the virtual machines have predefined minimum and maximum thresholds. If the memory of the VM increases the maximum threshold the MAC extends its memory, however, if it decreases the minimum threshold the MAC shrinks its memory with aim of more effective memory utilization. A computer with Intel processor and four-gigabyte dual ranked memory as hardware and VMware workstation as virtualization platform are used in this experiment. Also, five Linux OS is used including one for host-machine and four for guest machines. The results of the research show successful effective memory utilization.

(Xiao, et al., 2013) made a proactive research with virtual machines for cloud based environments which manages physical machines as well as virtual machines in order to use physical machines effectively. There are two main purpose of this study the first one of which is to prevent overload which may lead to the some kind of system failures and degradation of the performance. The other of which is saving energy by keeping the count of the active servers effectively which is also known as Green Computing. To handle these objectives some resource thresholds are defined on the system, which are named as hot and cold spots. The system collects the resources of both virtual machines and physical machines and predicts their future values with the algorithm that is Exponentially weighted moving Average (EWMA). The mathematical formula of the EWMA is given below:

$$E(t) = \alpha . E(t-1) + (1-\alpha) . O(t), 0 \leq \alpha \leq 1 \tag{3.1}$$

Where $E(t)$ and $O(t)$ are estimated and observed loads orderly at time $t$ orderly then $\alpha$ means a trade-off between stability and responsiveness.

After getting the prediction results, the system decides to the future resource status of the physical machine after revising the demanded resources of the virtual machines. If all the resources of PM will be below the cold spot, system understands that this PM will mostly be in idle state and migrates the VMs on it to another host. If the system decides that the resource usage of the PM will be above the hot spot, it searches some of the VMs to migrate away in order to avoid overload to keep general performance effectiveness. Another positive side of the study is Green Computing. The system can shut down the PMs which are in cold spot in order to save energy. 30 Dell PowerEdge blade servers which have Intel E5620 and 24 GB of RAM are used in the experiments.

Also, the hypervisor used is Xen 3.3 and the Operating System used is Linux 2.6.18. The results of the simulations are successful.

(Sommer, et al., 2016) presented another proactive study in cloud-based platforms by using Ensemble Forecasting. The main goal of this study is to prevent SLA violations by using the PMs in an effective manner. Thanks to the dynamic algorithm, Ensemble Forecasting, the system predicts the future workload of both guest and host machine and regulates the resources of VMs accordingly. In that process, if the host machine is categorized as the hot point which means the PM has the high possibility to get into the overload situation, the system starts to plan to migrate certain VMs from that host to a more proper one with more available resources. This approach results in more balanced usage of the host machines while preventing the overload issues resulting in less minimum SLA violations. In the experiments, eight hundreds hosts in which the first half is HP ProLiant ML110 G4 (Intel Xeon 3040, dual cores with 1860 MHz, 4 Ram) and the second half is HP ProLiant ML110 G5 (Intel Xeon 3075, dual cores with 2660 MHz, 4 GB Ram) is used as IaaS Environment ad CPU is used as selected resource to be predicted and revised according to the prediction results. Overall study has good results.

Green Computing has a special place for (Lu, et al., 2016). They made a proactive research on predicting the future workload of server resources in order to scale them accordingly for eliminating undesirable energy consumption while preventing the SLA violations. They proposed Rand Variable Learning Rate Backpropagation Neural Network (RVLBPNN) based on Backpropagation Neural Network (BNN) algorithm. Matlab is used for the experiments where 28 days of Google usage data is used. The results showed that RVLBPNN has better prediction accuracy compared to the HMM and Naive Bayes Classifier models.

(Erradi & Kholidy, 2017) stated a new proactive method called Data Driven Hybrid Prediction Approach (DDHPA) which consists of Multiple Support Vector Regression (MSVR) and Autoregressive Integrated Moving Average (ARIMA) models in order to forecast changing consumer resource usage which are CPU, memory and storage more accurately. The main goal of this study is to develop a model which has not any kind of shortcomings rivals have so as to predict more accurate results in both short and long periods even in the most fluctuating cases. The stated model DDHPA has three main modules one of which is feature selection in which the critical properties for the future prediction are selected resulting in more accurate results as well as better memory performance by reducing the requirement, another of which is training and parameter estimation where the Particle Swarm Optimization (PSO) is used to select appropriate features to be used in both ARIMA and MSVR for better prediction, the other of which is testing and validation in which Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) is used. Experiments showed successful results.

(Calheiros, et al., 2014) studied on realization workload prediction with the help of Autoregressive Integrated Moving Average (ARIMA) which is also proactive method. The main purpose of the study is to state a good prediction module for the auto-scaling and the evaluation of the system effectiveness. The experiment data is get from the Wikimedia Foundation and only the English Wikipedia requests are taken into account. The simulation environment is a data center with 1.000 PMs which have eight cores and 16 GB RAM. Results of the experiment showed that the prediction accuracy reached to the levels of 91 percent with minimal performance reduction.

Another proactive research is done by (Vazquez, et al., 2015) about Dynamic Resource Provisioning of Cloud Data Centers. The philosophy of behind of this study is that reactive provisioning cannot meet the requirements of the changing customer demands in time without violating the SLA. Thus, a number of prediction modules are evaluated from the performance and applicability points of views because the most accurate prediction is the key factor in Dynamic Resource Provisioning. Autoregressive Model is the first prediction model that is used as first-order AR(1). Moving Average is another forecasting model which is also used as first-order moving average (MA(1)). Simple exponential Smoothing is the other prediction algorithm. Double exponential smoothing is the fourth technique assessed which is more complicated version of the simple exponential smoothing. ETS is the fifth algorithm evaluated in this study. ARIMA is the sixth algorithm including autoregressive (AR(p)), Integrated (I(q)) and moving average (MA(d)) parts. Neural Network Regression is the final method used in that research which composed of neurons and weights like human mind and three layer structure which are input, hidden and output. For the prediction evaluation again several methods are tested which are Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Scaled Error (MASE). Three tests are done for the assessment of prediction methods which are Out of sample forecasting error in which the model is trained with the past data to make better predictions, Rolling forecast origin cross validation which does not have a historical data to train but it learns from data itself, Training set length analysis in which train set slid forward to a couple of times and the predictions are done. R programming language is used for the experiments which are done on a computer with Intel Core i7-4500U CPU and 8 GB RAM. As the dataset Intel Netbatch logs and Google cluster data is used. The results of the studies showed that although the performance of the prediction method varies based on several factors such as data and time, first order autoregressive and neural network auto regression is better than the others.

There are a number of researches done on the Dynamic Resource Provisioning and the several of them are listed above. Although certain studies include reactive kind of methods, general tendency is towards to the proactive ones as expected. As seen from the proven results proactive methods based on prediction are much more effective than reactive ones because of the fact that they reduce overprovisioning as well as under provisioning resulting in monetary save, energy save  and better Quality of Service without process overhead. Therefore, this research is designed as proactive.

# 4. DYNAMIC RESOURCE PROVISIONING

A number of experiments have been done in the scope of this research. This section will explain all of them step by step. The overview of this section is designed as follows: System architecture and procedure is given in the first place. Secondly, data sets getting from Radore Data Center through the Zabbix Monitoring Engine will be explained. Then, lists of experiments will start which are implementation of LSTM, implementation of ARIMA, implementation of Dynamics Resource Provisioning on VMware, implementation of Dynamic Resource Provisioning on OnApp, evaluation of these provisioning issues, implementation of creating VM instance on VMware, implementation of creating VM instance on OnApp and evaluation of these instance creation processes.

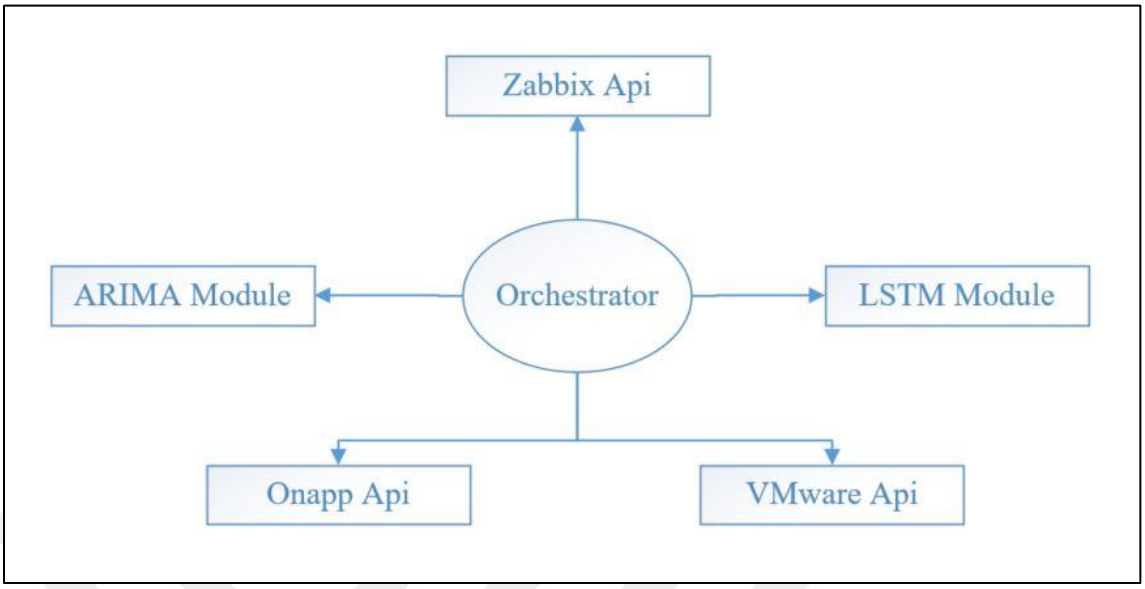
## 4.1 SYSTEM ARCHITECTURE AND PROCEDURE

The architecture of the system is composed of six software modules which are as the following:

i. Zabbix data collection module
ii. LSTM module
iii. ARIMA module
iv. Vmware module
v. OnApp module
vi. General orchestraton module

The overall diagram is given in *Figure 4.1* below:

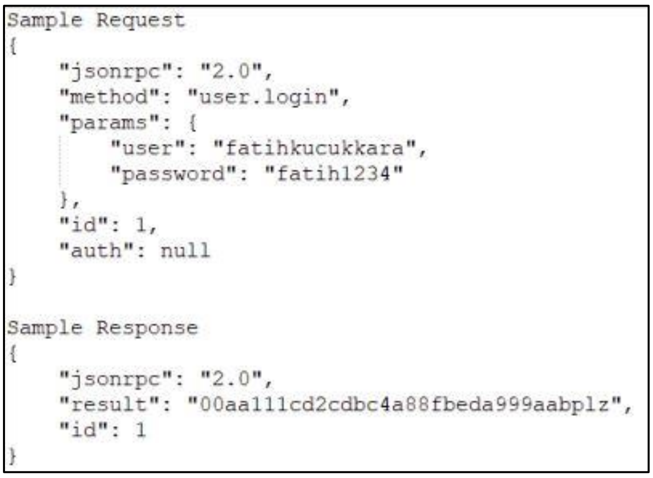
**Figure 4.1: The software architecture**



### 4.1.1 Zabbix Data Collection Module

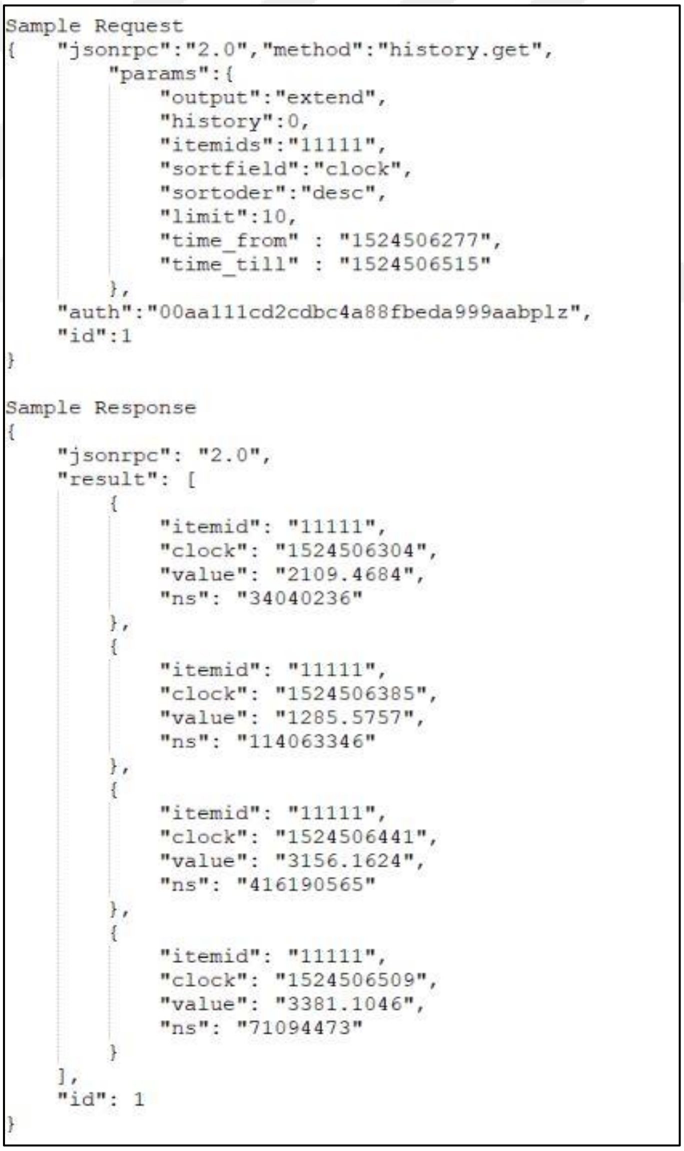
Zabbix Data Collection Module is an N-Tier web API which is developed by using C# programming language on the framework, Asp.Net 4.6.2. The job of the Zabbix Data Collector is to get the specified monitoring data from the Central Zabbix Server. This module is an implementation of the Zabbix "history.get" command stated in the Zabbix Documentation 3.0 (Zabbix, 2018). In order to consume Zabbix API, getting API key is required. The sample JSON formatted request and response prepared to be send to the zabbix server is stated in *Figure 4.2*. After getting the API key, the required command can be sent. The sample request and response command prepared to be send to the zabbix server is given in the *Figure 4.3*.

**Figure 4.2: Sample request-response for API key**



```
Sample Request
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "user": "fatihkucukkara",
        "password": "fatih1234"
    },
    "id": 1,
    "auth": null
}

Sample Response
{
    "jsonrpc": "2.0",
    "result": "00aa111cd2cdbc4a88fbeda999aabplz",
    "id": 1
}
```

**Figure 4.3: Sample request- response for history**



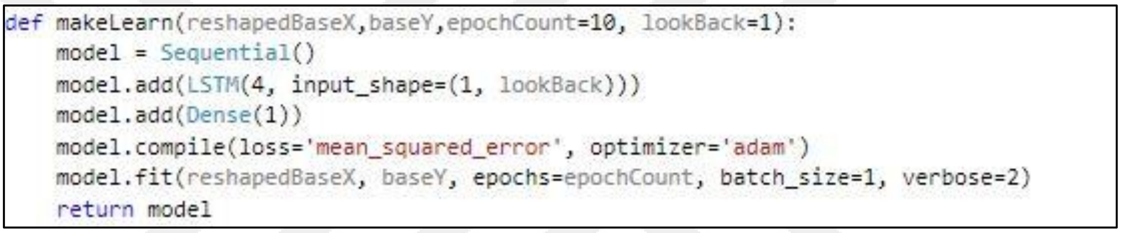
```
Sample Request
{   "jsonrpc":"2.0","method":"history.get",
        "params":{
            "output":"extend",
            "history":0,
            "itemids":"11111",
            "sortfield":"clock",
            "sortoder":"desc",
            "limit":10,
            "time_from" : "1524506277",
            "time_till" : "1524506515"
        },
    "auth":"00aa111cd2cdbc4a88fbeda999aabplz",
    "id":1
}

Sample Response
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "11111",
            "clock": "1524506304",
            "value": "2109.4684",
            "ns": "34040236"
        },
        {
            "itemid": "11111",
            "clock": "1524506385",
            "value": "1285.5757",
            "ns": "114063346"
        },
        {
            "itemid": "11111",
            "clock": "1524506441",
            "value": "3156.1624",
            "ns": "416190565"
        },
        {
            "itemid": "11111",
            "clock": "1524506509",
            "value": "3381.1046",
            "ns": "71094473"
        }
    ],
    "id": 1
}
```

### 4.1.2 LSTM Module

LSTM Module is a software library which is developed in python 3.6. This module gets the monitored data by the Zabbix Data Collection Module and transforms it to range 0-1 in order to eliminate noise. Then, data is divided into two sub-groups which are train and test. After some modelling work, train data is used to create LSTM model for predictions. This module is prepared according to the “Out Of Sample” methodology where the system is trained with the learning data and prediction is made with the test data.

The core of the module where learning occurs is given in *Figure 4.4*.

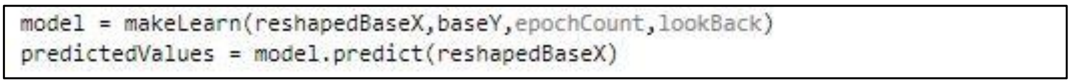
**Figure 4.4: The lstm module train part**



```
def makeLearn(reshapedBaseX,baseY,epochCount=10, lookBack=1):
    model = Sequential()
    model.add(LSTM(4, input_shape=(1, lookBack)))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')
    model.fit(reshapedBaseX, baseY, epochs=epochCount, batch_size=1, verbose=2)
    return model
```

The main class LSTM is imported from the Keras Deep Learning Library with the version 2.1.2. In that method, reshapedBaseX refers to input, baseY refers to real output, epochCount refers to iteration number and lookback refers to a number of previous values that are used for prediction.  After learning part, the prediction is made through the test data like in *Figure 4.5*.

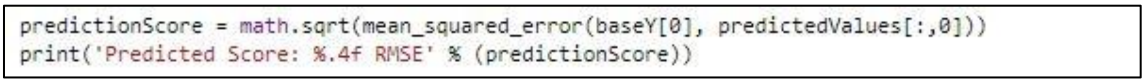
**Figure 4.5: The lstm module prediction part**



```
model = makeLearn(reshapedBaseX,baseY,epochCount,lookBack)
predictedValues = model.predict(reshapedBaseX)
```

After the prediction, RMSE is implemented as the error checker method like *Figure 4.6*.

**Figure 4.6: The lstm module error checking part**

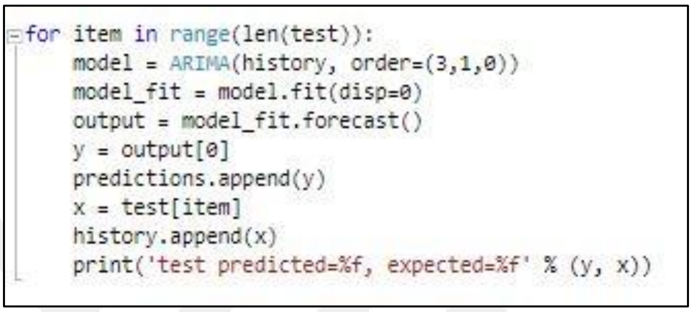


```
predictionScore = math.sqrt(mean_squared_error(baseY[0], predictedValues[:,0]))
print('Predicted Score: %.4f RMSE' % (predictionScore))
```

### 4.1.3 ARIMA Module

ARIMA Module is also a software library which is developed in python 3.6. This module gets the monitored data provided by the Zabbix Data Collection Module, scales it to the range 0-1 and then divides it into two sub-data as train and test after some formatting works. After this categorization, train data is used to create a model and test data is used to test prediction quality like LSTM Module. The difference is the methodology for the prediction. ARIMA Module is prepared with the method of Rolling Forecast in which every predicted result is used to create a new model for next predictions. The core of the module is given in *Figure 4.7*.

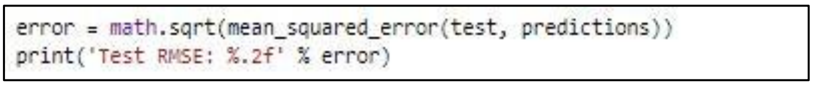
**Figure 4.7: The arima module prediction part**



```
for item in range(len(test)):
    model = ARIMA(history, order=(3,1,0))
    model_fit = model.fit(disp=0)
    output = model_fit.forecast()
    y = output[0]
    predictions.append(y)
    x = test[item]
    history.append(x)
    print('test predicted=%f, expected=%f' % (y, x))
```

The main class seen in *Figure 4.*4 that ARIMA is imported from STATSMODELS which is the version 0.8.0. The key part of this code is the parameters of the ARIMA which is 3, 1, 0. These parameters are actually the components of the ARIMA that are p (the lag), d(the difference), q(moving average) mentioned in Section 2.7. Another important point is error detection. RMSE is implemented like the LSTM Module, which is seen from the *Figure 4.8.*

**Figure 4.8: The arima module error checking part**



```
error = math.sqrt(mean_squared_error(test, predictions))
print('Test RMSE: %.2f' % error)
```

### 4.1.4 VMware Module

VMware Module which enables us to create, get, update and edit VMs on the VMware is composed of two web APIs developed with C# programming language and a powershell script. The script is the core part of communicating with VMware. There is a low level API over the script which manages the requests and responses with the script. Also, there is a high level API which manages requests coming from outside and responses coming from the low-level API, which is a kind of wrapper API meeting the requests for different kinds of cloud infrastructure platforms which are VMware

Enterprise Cloud, VMware Cloud and OnApp Cloud and dispatch them to the related low level APIs.

### 4.1.5 OnApp Module

OnApp Module is a part of the wrapper API mentioned in the 4.1.3 which enables us to do certain VM functionalities such as create, get, update and delete. This module communicates with the OnApp low-level API that releases the documentation 5.0 (Pushkar & Dubno, 2018).

### 4.1.6 General Orchestration Module

General Orchestration Module is a python library where the synchronization of the API calls is done. This module calls the APIs, gets the results and sends them to other modules after doing required formatting tasks. To prevent any conflict or a tracking problem, the high level APIs do not call each other. As seen from the *Figure 4.1*, the orchestrator that is at the center collects the results and decides what to do next according to the business rules mentioned in the Section 4.1.7.

### 4.1.7 Procedure

In that kind of architecture, the general orchestration module (OM) must be in an outside controller server. OM collects the required monitored data from the VMs and PMs in certain time periods according to the type of business rule. Then, OM sends the collected monitoring data to the prediction module and gets the prediction results. After that OM deciding what to do by checking business rules which includes data check periods mentioned above, minimum and maximum resource amounts for resource types for hot resizing, hot and cold spots for resource types to decide to increase the amount of resource or decrease it orderly, hot and cold spots for host machines to transfer VMs for the save of performance or for the safe of energy consumption by closing the host orderly and so on. After the decision point, OM gives the action command to related API. In the simulation of this study, hot resize and creation of VM are covered however, the system supports all the rules and more mentioned above.

## 4.2 DATASET

Datasets are got from the Radore Data Center where a number of services such as hosting, domain, dedicated servers, cloud servers and disaster services are given. In that study, five datasets are got from different kind of VMs from different business domains. The values of the datasets are collected per minute by the Zabbix Server.

### 4.2.1 Dataset 1

This dataset is got from a VM which is used for internal purposes. The system runs on the VMware infrastructure and has provisioned with 16 GB memory.

**Figure 4.9: Dataset 1 free memory 5-minute time range**

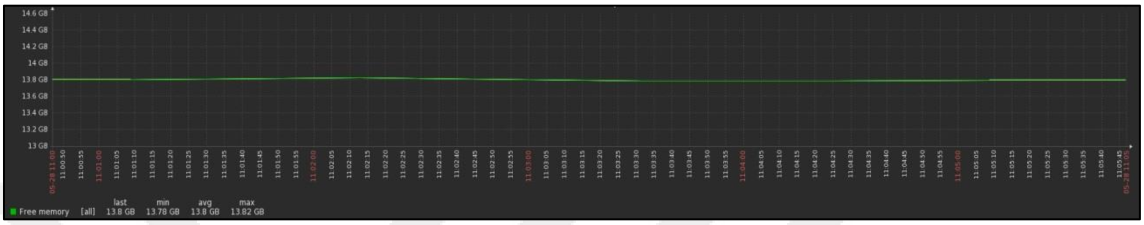


*Figure 4.9* shows 5-minute time range free memory graph. The minimum free memory amount 13.78 GB and the maximum free memory amount is 13.82 GB.

**Figure 4.10: Dataset 1 free memory 1-hour time range**

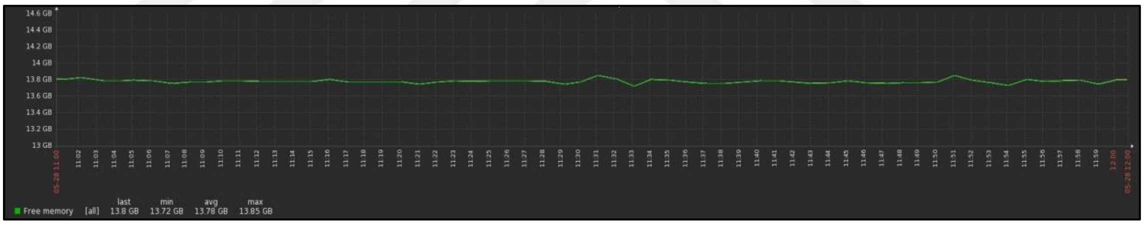


*Figure 4.10* shows 1-Hour time range graph. The minimum free memory amount 13.72 GB and the maximum free memory amount is 13.85 GB.

**Figure 4.11: Dataset 1 free memory 1-day time range**

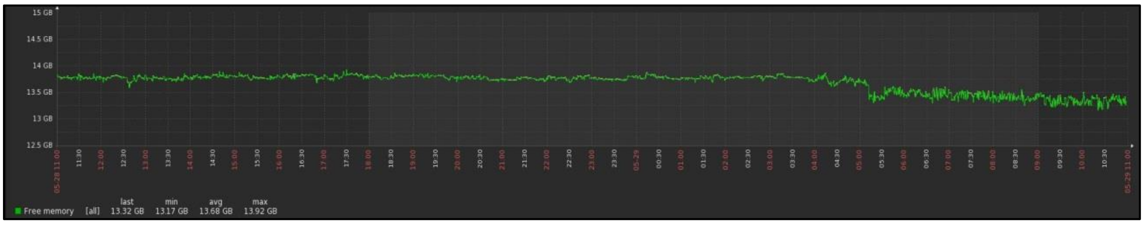


*Figure 4.11* shows 1-Day time range graph. The minimum free memory amount 13.17 GB and the maximum free memory amount is 13.92 GB.

**Figure 4.12: Dataset 1 free memory 7-day time range**

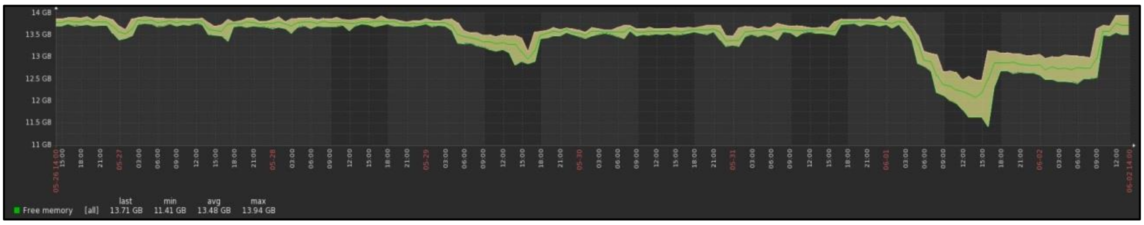


*Figure 4.12* shows 7-Day time range graph. The minimum free memory amount 11.41 GB and the maximum free memory amount is 13.94 GB.

**Figure 4.13: Dataset 1 free memory 1-month time range**

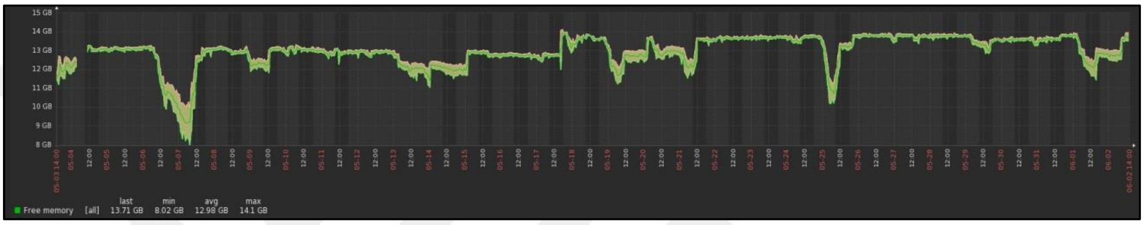


*Figure 4.13* shows 1-Month time range graph. The minimum free memory amount 8.02 GB and the maximum free memory amount is 14.1 GB.

**Figure 4.14: Dataset 1 free memory 3-month time range**

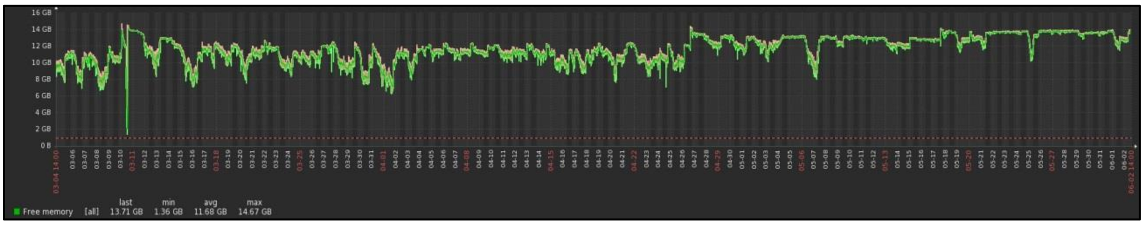


*Figure 4.14* shows 3-Month time range graph. The minimum free memory amount 13.71 GB and the maximum free memory amount is 14.67 GB.

We see that from the results, the free memory amount is about 12 GB. The company has kept the memory amount high to prevent starvation in peak times, leading to the memory wastage.

### 4.2.2 Dataset 2

This dataset is get from a VM which is used for internal purposes. The system runs on the VMware infrastructure and has provisioned with 16 GB memory.

**Figure 4.15: Dataset 2 free memory 5-minute time range**

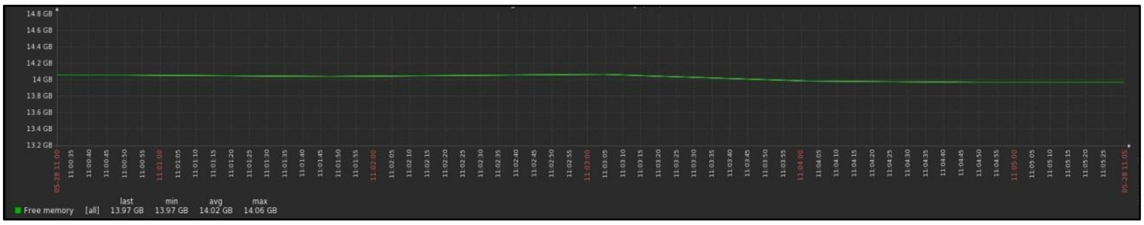


*Figure 4.15* shows 5-Minute time range graph. The minimum free memory amount 13.97 GB and the maximum free memory amount is 14.06 GB.

**Figure 4.16: Dataset 2 free memory 1-hour time range**

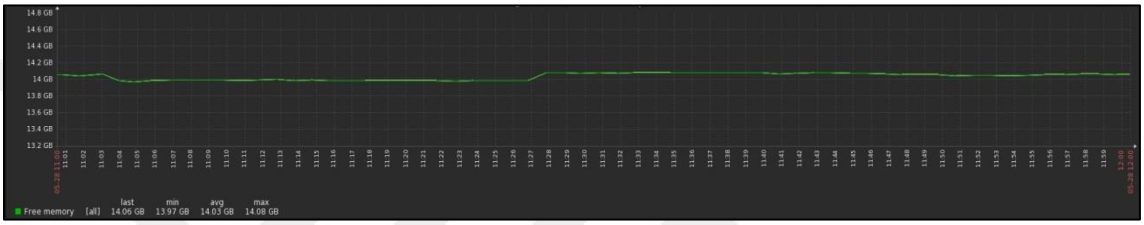


Figure 4.16 shows 1-Hour time range graph. The minimum free memory amount 13.97 GB and the maximum free memory amount is 14.08 GB.

**Figure 4.17: Dataset 2 free memory 1-day time range**

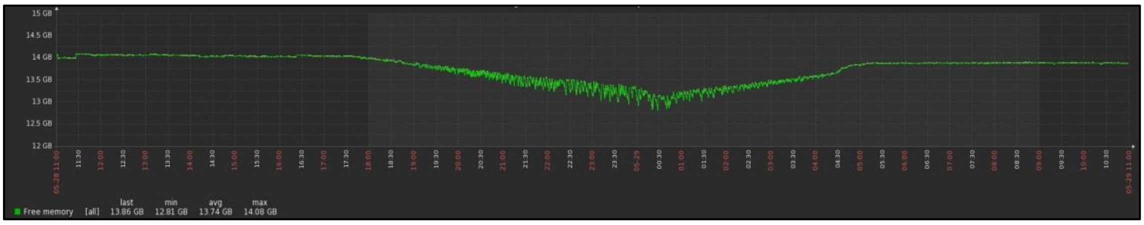


*Figure 4.17* shows 1-Day time range graph. The minimum free memory amount 12.81 GB and the maximum free memory amount is 14.08 GB.

**Figure 4.18: Dataset 2 free memory 7-day time range**

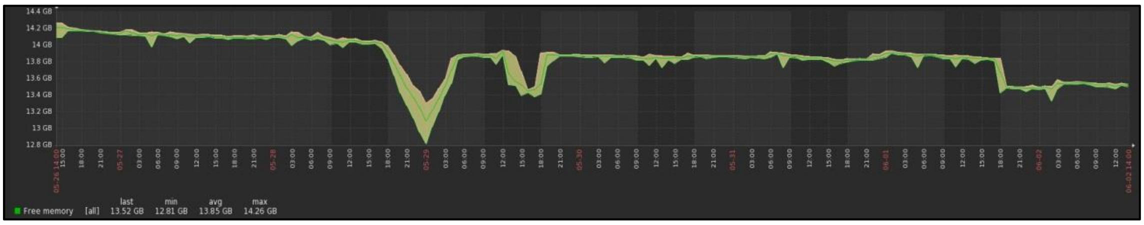


*Figure 4.18* shows 7-Day time range graph. The minimum free memory amount 12.81 GB and the maximum free memory amount is 14.26 GB.

**Figure 4.19: Dataset 2 free memory 1-month time range**

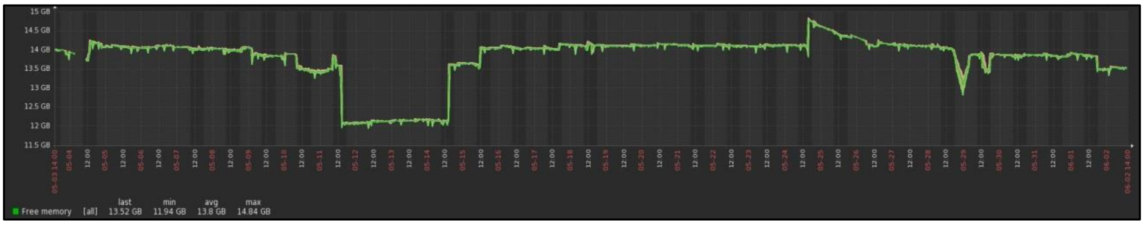


Figure 4.19 shows 1-Month time range graph. The minimum free memory amount 11.94 GB and the maximum free memory amount is 14.84 GB.

**Figure 4.20: Dataset 2 free memory 3-month time range**

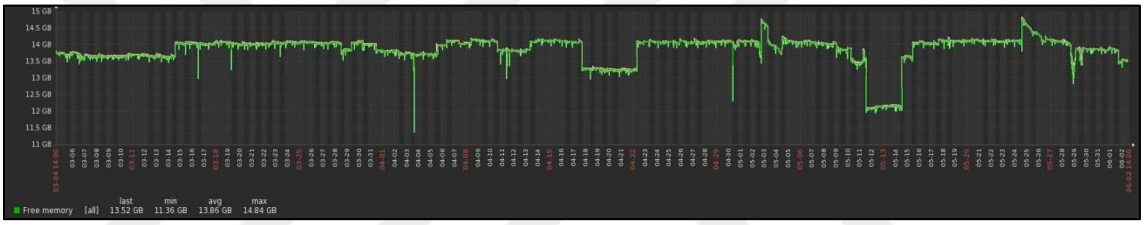


*Figure 4.20* shows 1-Month time range graph. The minimum free memory amount 11.36 GB and the maximum free memory amount is 14.84 GB.

The results show that the free memory amount of this server has never fallen down below the 11 GB, which is another sample of resource wastage.

### 4.2.3 Dataset 3

This real dataset is get from a VM. The system runs on the VMware infrastructure and has provisioned with 32 GB memory.

**Figure 4.21: Dataset 3 free memory 5-minute time range**

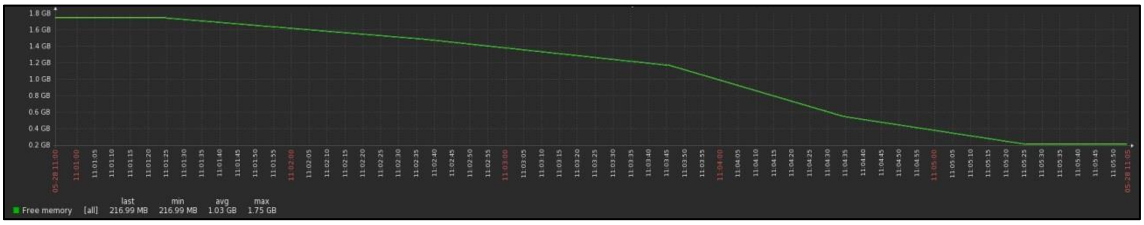


*Figure 4.21* shows 5-Minute time range graph. The minimum free memory amount 216.99 MB and the maximum free memory amount is 1.75 GB.

**Figure 4.22: Dataset 3 free memory 1 hour time range**

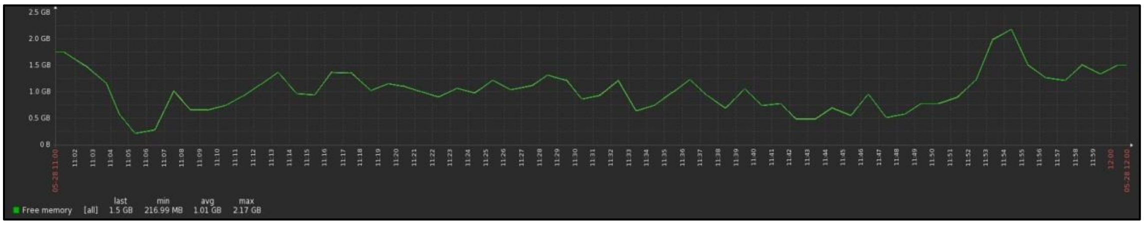


*Figure 4.23* shows 1-Hour time range graph. The minimum free memory amount 216.99 MB and the maximum free memory amount is 2.17 GB.

**Figure 4.23: Dataset 3 free memory 1-day time range**

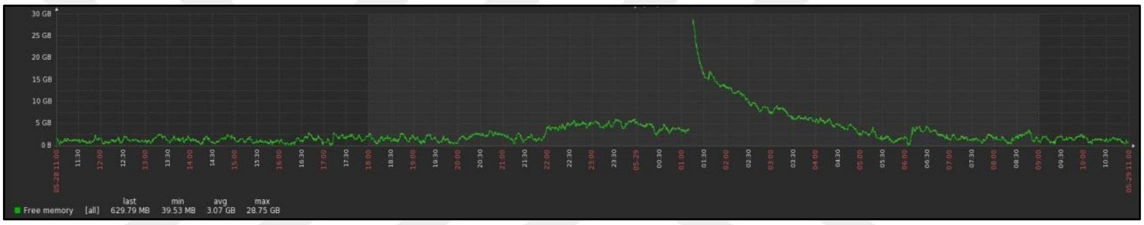


*Figure 4.23* shows 1-Day time range graph. The minimum free memory amount 39.53 MB and the maximum free memory amount is 28.75 GB.

**Figure 4.24: Dataset 3 free memory 7-day time range**

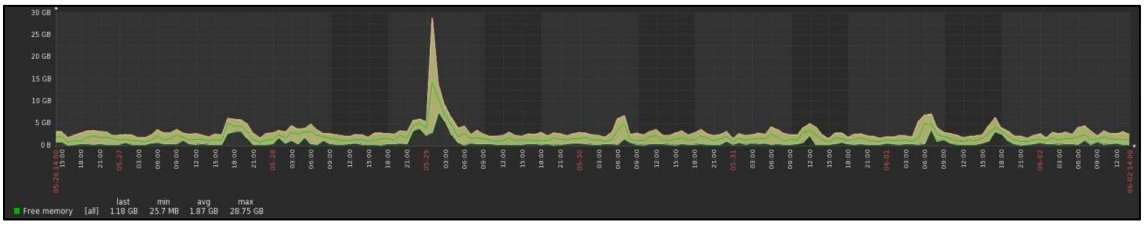


*Figure 4.24* shows 7-Day time range graph. The minimum free memory amount 25.7 MB and the maximum free memory amount is 28.75 GB.

**Figure 4.25: Dataset 3 free memory 1-month time range**

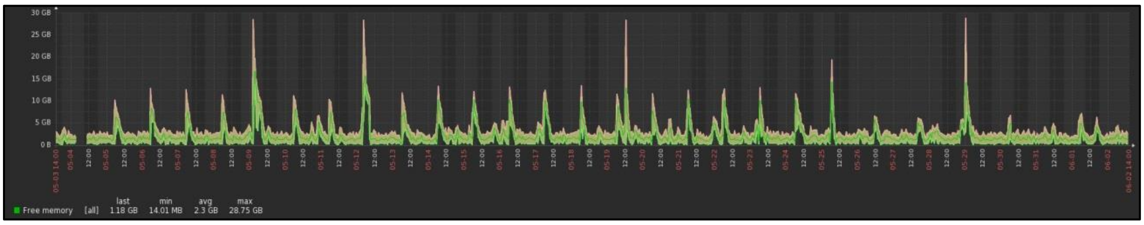


*Figure 4.25* shows 1-Month time range graph. The minimum free memory amount 14.01 MB and the maximum free memory amount is 28.75 GB. As seen from the graphics, there are some regular local peaks as well as global ones.

**Figure 4.26: Dataset 3 free memory 3-month time range**

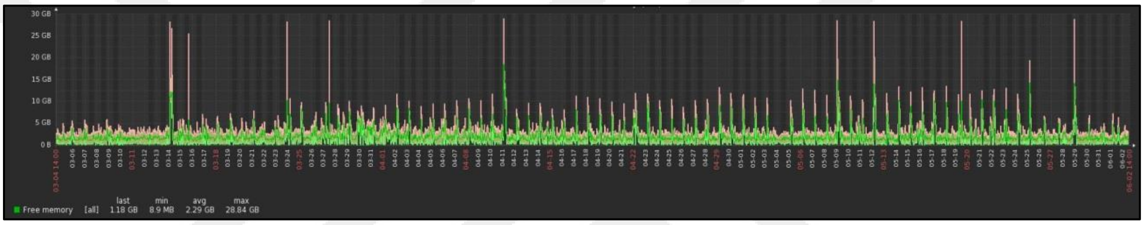


*Figure 4.26* shows 1-Month time range graph. The minimum free memory amount 8.9 MB and the maximum free memory amount is 28.84 GB.

The results show that the resource usage of that VM has been mostly high, leading to the overload and performance degradation at the end. Also we can see certain paths that are regular increases and decreases, which can be handled with the predictive memory allocation effectively.

### 4.2.4 Dataset 4

This real dataset is get from a VM which is used for internal purposes. The system runs on the VMware infrastructure and has provisioned with 64 GB memory.

**Figure 4.27: Dataset 4 free memory 5-minute time range**

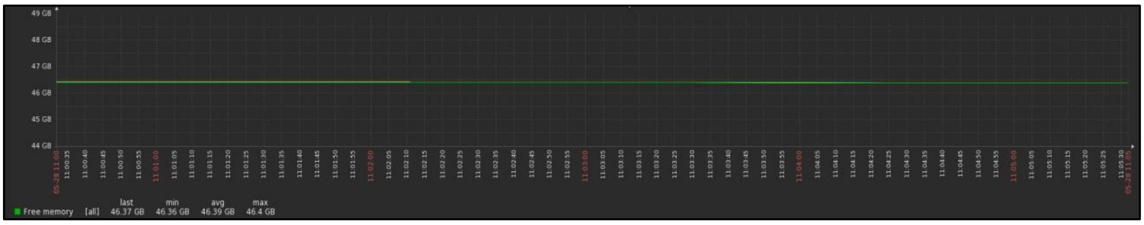


*Figure 4.27* shows 5-Minute time range graph. The minimum free memory amount 46.36 GB and the maximum free memory amount is 46.4 GB.

**Figure 4.28: Dataset 4 free memory 1-hour time range**

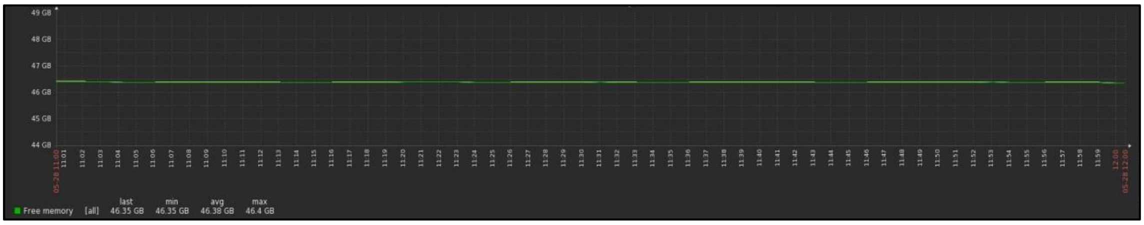


*Figure 4.28* shows 1-Hour time range graph. The minimum free memory amount 46.35 GB and the maximum free memory amount is 46.4 GB.

**Figure 4.29: Dataset 4 free memory 1-day time range**

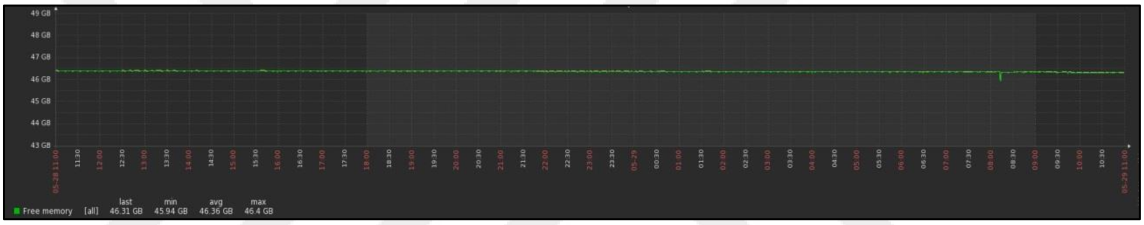


*Figure 4.29* shows 1-Day time range graph. The minimum free memory amount 45.94 GB and the maximum free memory amount is 46.4 GB.

**Figure 4.30: Dataset 4 free memory 7-day time range**

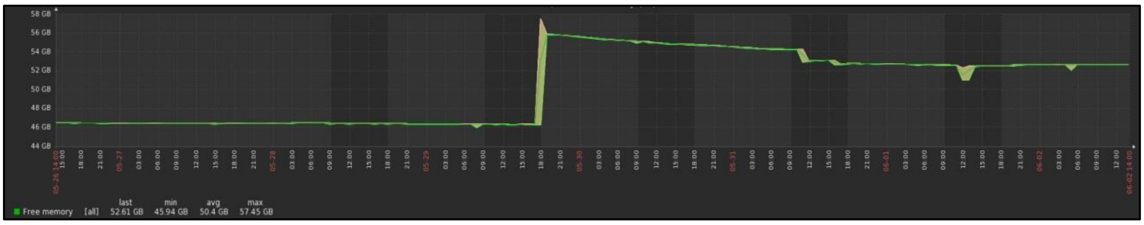


*Figure 4.30* shows 7-Day time range graph. The minimum free memory amount 45.94 GB and the maximum free memory amount is 57.45 GB.

**Figure 4.31: Dataset 4 free memory 1-month time range**

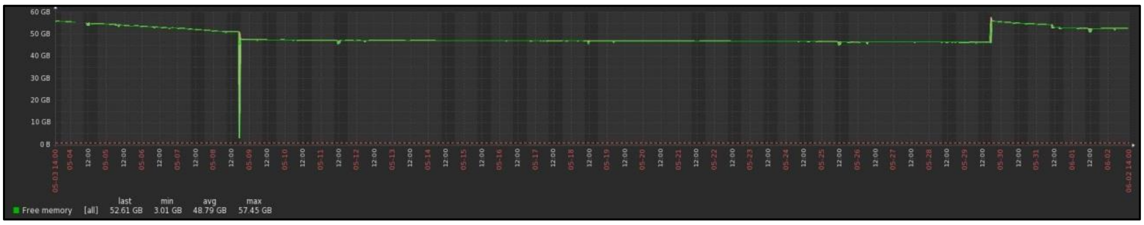


*Figure 4.31* shows 1-Month time range graph. The minimum free memory amount 3.01 GB and the maximum free memory amount is 57.45 GB.

**Figure 4.32:Dataset 4 free memory 3-month time range**

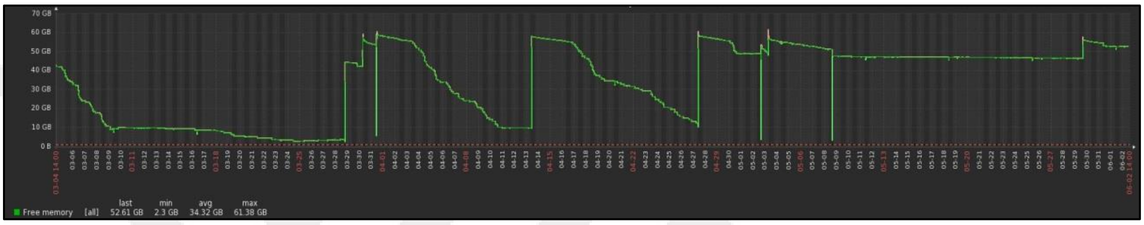


*Figure 4.32* shows 1-Month time range graph. The minimum free memory amount 2.3 GB and the maximum free memory amount is 61.38 GB.

64 GB provisioned memory refers that this is a busy VM but the graphs show that most of the time the resource has not been used effectively. As it can see from the *Figure 4.30*, *Figure 4.31* and *Figure 4.32*, there are instant sharp increases from even 2.3 GB to 61.38 GB which cannot be handled with any gradual increase methodology without degradation.

### 4.2.5 Dataset 5

This real dataset is get from a VM. The system runs on the VMware infrastructure and has provisioned with 64 GB memory.

**Figure 4.33: Dataset 4 free memory 5-minute time range**

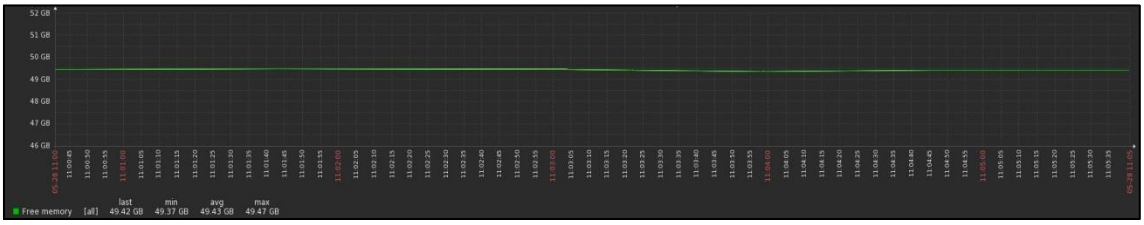


*Figure 4.33* shows 5-Minute time range graph. The minimum free memory amount 49.37 GB and the maximum free memory amount is 49.47 GB.

Figure 4.34: Dataset 4 free memory 1-hour time range

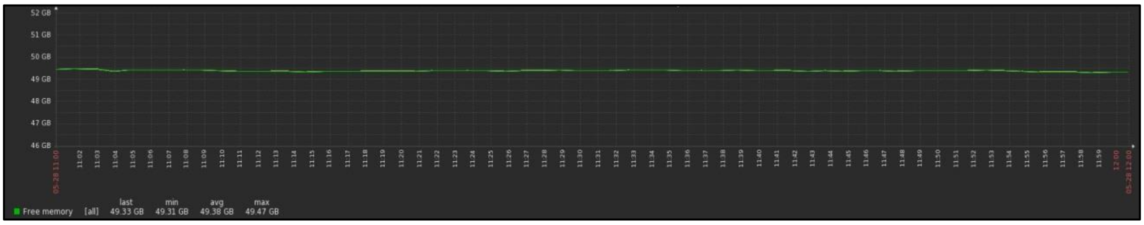


*Figure 4.34* shows 1-Hour time range graph. The minimum free memory amount 49.31 GB and the maximum free memory amount is 49.47 GB.

Figure 4.35: Dataset 4 free memory 1-day time range

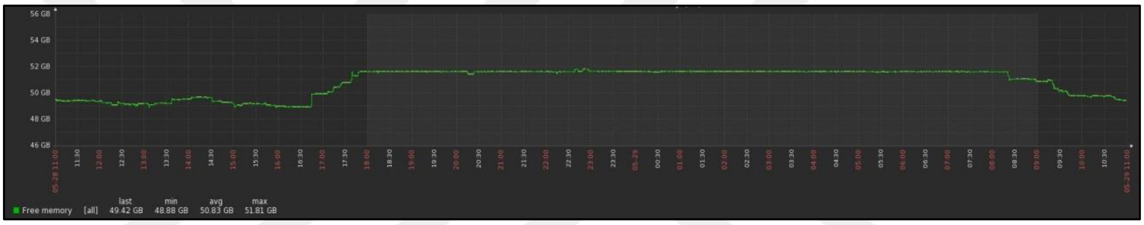


*Figure 4.35* shows 1-Day time range graph. The minimum free memory amount 48.88 GB and the maximum free memory amount is 51.81 GB.

Figure 4.36: Dataset 4 free memory 7-day time range

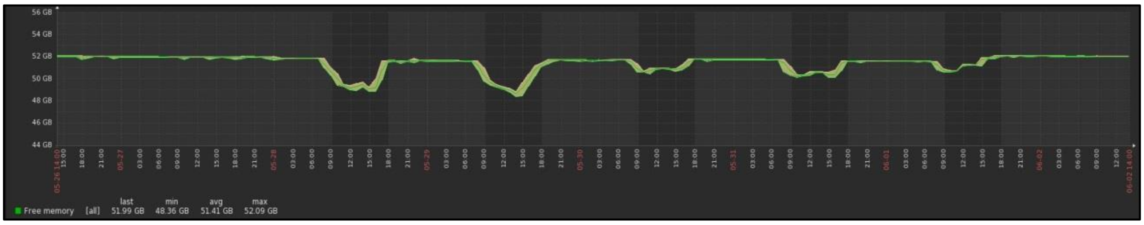


*Figure 4.36* shows 7-Day time range graph. The minimum free memory amount 48.36 GB and the maximum free memory amount is 52.09 GB.

Figure 4.37: Dataset 4 free memory 1-month time range

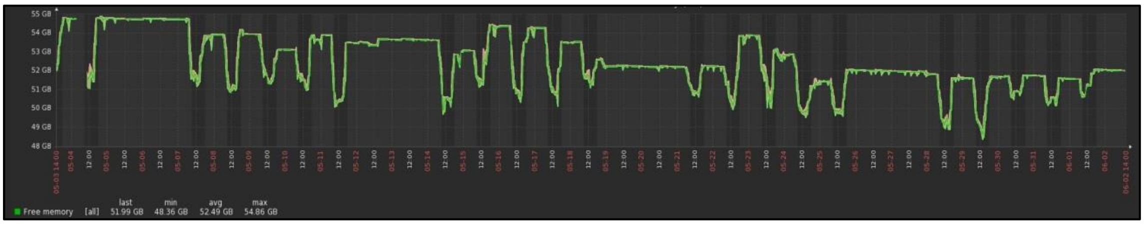


*Figure 4.37* shows 1-Month time range graph. The minimum free memory amount 48.36 GB and the maximum free memory amount is 54.86 GB.

**Figure 4.38: Dataset 4 free memory 3-month time range**

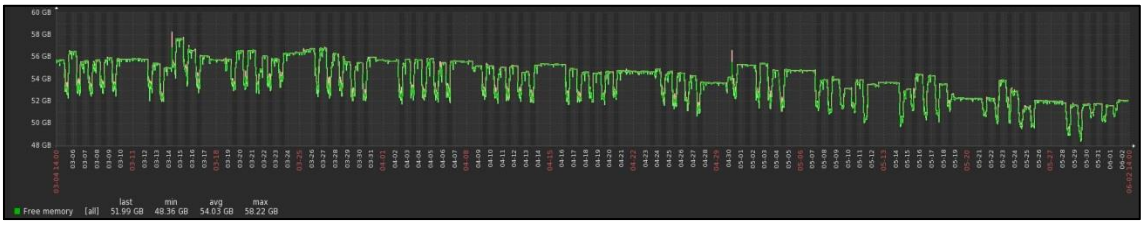


*Figure 4.38* shows 3-Month time range graph. The minimum free memory amount 48.36 GB and the maximum free memory amount is 58.22 GB.

Although there have seen some regular fluctuations at certain periods, the available memory has not fallen down below the 48 GB, which is a tragic resource wastage. The system again setup with very high resources to prevent starvation resulting in a great resource wastage.

## 4.3 TARGET POPULATION

The main population that can use this system is Data Centers but other companies or individuals can easily use it in their cloud systems. This system is capable of doing a number of different tasks based on the business rules mentioned in Section 4.1.7 however, prediction is the most important part of the system and predictive vertical scaling without reboot is the most effective part. Thus, medium-sized companies that have fluctuating resource demands are the group who can benefit most.

## 4.4 TESTS

In that part, datasets described in Section 4 are used in predictive tests with the algorithms LSTM and ARIMA. Then, performances and capabilities of the VMware and OnApp Cloud Platforms are shown. Toshiba i7-4720 HQ CPU @ 2.60 GHz, 16 GB Ram with Windows 10 x64 OS is the computer in which all tests are run.

### 4.4.1 LSTM Tests

The first two tests have been done to determine the best LSTM configuration, which are done with 1-Hour data to save time. The longer time periods require more computational time. Other tests are done to test the algorithm. Again, due to the long-time requirements of 1-Week and the longer time periods, 1-Day time period is used in tests. However, to get more realistic results, q sequence of 1-Day periods are tested and their average is calculated as result.

#### 4.4.1.1 LSTM Test 1

**Table 4.1: LSTM test 1 configuration**

| **Dataset** | **Dataset 1** |
|---|---|
| **Data Period – Data Count** | 1 hour – 60 |
| **Lag** | 1 |
| **Epoch** | 100 |
| **Test RMSE** | **17805380.64 byte** |

*Table 4.1* gives certain information about the data and the algorithm configuration. 1-Hour time period data containing 60 time series data is used with the LSTM configuration 1 Lag, 100 Epoch.

**Figure 4.39: LSTM test 1**

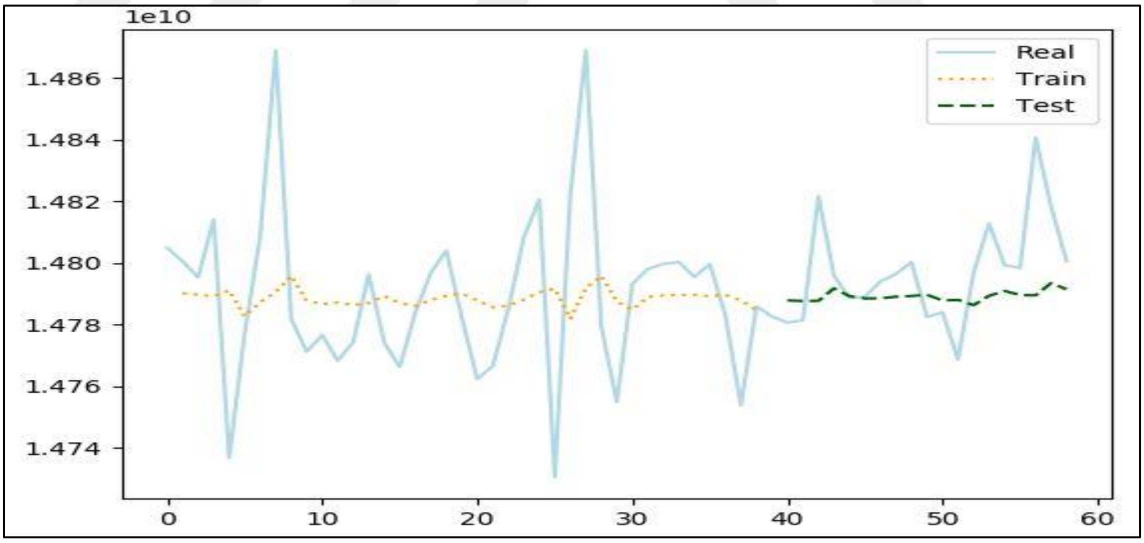


*Figure 4.39* shows the result of the test that is done with the configuration mentioned *Table 4.1*. As seen from the legend, blue colour means the real time series data, the orange one means the train prediction and the green one means the test prediction.

#### 4.4.1.2 LSTM Test 2

**Table 4.2: LSTM test 2 configuration**

| **Dataset** | **Dataset 1** |
|---|---|
| **Data Period – Data Count** | 1 hour – 60 |
| **Lag** | 3 |
| **Epoch** | 300 |
| **Test RMSE** | **25522266.62 byte** |

*Table 4.2* gives certain information about the data and the algorithm configuration. 1-Hour time period data containing 60 time series data is used with the LSTM configuration 3 Lag, 300 Epoch.

**Figure 4.40: LSTM test 2**

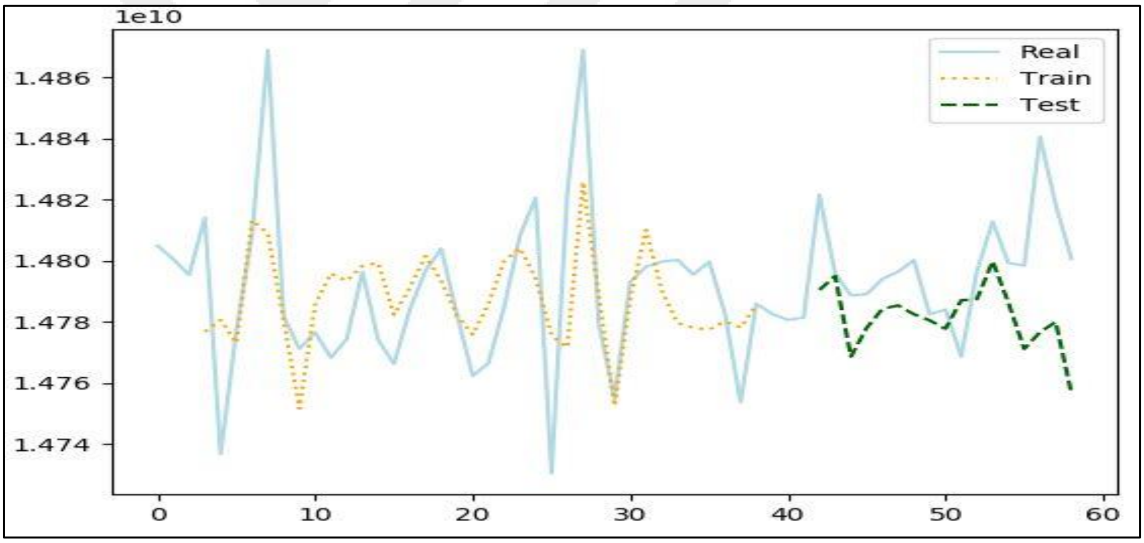


*Figure 4.40* shows the result of the test that is done with the configuration mentioned *Table 4.2*. As seen from the legend, blue colour means the real time series data, the orange one means the train prediction and the green one means the test prediction.

#### 4.4.1.3 LSTM Test 3

**Table 4.3: LSTM test 3**

| | **Dataset** | **Time Period** | **Data Count** | **Lag** | **Epoch** | **RMSE (byte)** |
|---|---|---|---|---|---|---|
| **1st Day** | 1 | 1-Day | 1440 | 1 | 100 | 50166863.22 |
| **2nd Day** | 1 | 1-Day | 1440 | 1 | 100 | 31967716.53 |
| **3rd Day** | 1 | 1-Day | 1437 | 1 | 100 | 141938957.13 |
| **4th Day** | 1 | 1-Day | 1440 | 1 | 100 | 25037008.44 |
| **5th Day** | 1 | 1-Day | 1440 | 1 | 100 | 33940948.52 |
| **6th Day** | 1 | 1-Day | 1440 | 1 | 100 | 828152316.86 |
| **7th Day** | 1 | 1-Day | 1440 | 1 | 100 | 71835979.85 |
| **Average** | | | | | | **169005684.36** |

*Table 4.3* informs us about a sequence of 1-Day tests which are done with the configuration 1 Lag and 100 Epoch. The data counts are given in the table. The results are got with the unit of byte.

#### 4.4.1.4 LSTM Test 4

**Table 4.4: LSTM test 4**

| | **Dataset** | **Time Period** | **Data Count** | **Lag** | **Epoch** | **RMSE (byte)** |
|---|---|---|---|---|---|---|
| **1st Day** | 2 | 1-Day | 1440 | 1 | 100 | 8370198.91 |
| **2nd Day** | 2 | 1-Day | 1440 | 1 | 100 | 21787944.46 |
| **3rd Day** | 2 | 1-Day | 1436 | 1 | 100 | 87189685.89 |
| **4th Day** | 2 | 1-Day | 1440 | 1 | 100 | 13900629.12 |
| **5th Day** | 2 | 1-Day | 1440 | 1 | 100 | 12357307.05 |
| **6th Day** | 2 | 1-Day | 1440 | 1 | 100 | 19850412.58 |
| **7th Day** | 2 | 1-Day | 1440 | 1 | 100 | 7760822.95 |
| **Average** | | | | | | **24459571.56** |

*Table 4.4* informs us about a sequence of 1-Day tests which are done with the configuration 1 Lag and 100 Epoch. The data counts are given in the table. The results are got with the unit of byte.

### 4.4.1.5 LSTM Test 5

**Table 4.5: LSTM test 5**

| | **Dataset** | **Time Period** | **Data Count** | **Lag** | **Epoch** | **RMSE (byte)** |
|---|---|---|---|---|---|---|
| **1st Day** | 3 | 1-Day | 1440 | 1 | 100 | 346992537.02 |
| **2nd Day** | 3 | 1-Day | 1436 | 1 | 100 | 421079285.17 |
| **3rd Day** | 3 | 1-Day | 1436 | 1 | 100 | 350588031.20 |
| **4th Day** | 3 | 1-Day | 1440 | 1 | 100 | 386730667.93 |
| **5th Day** | 3 | 1-Day | 1440 | 1 | 100 | 336668935.92 |
| **6th Day** | 3 | 1-Day | 1440 | 1 | 100 | 336944478.25 |
| **7th Day** | 3 | 1-Day | 1440 | 1 | 100 | 327611939.50 |
| **Average** | | | | | | **358087982.14** |

*Table 4.5* informs us about a sequence of 1-Day tests which are done with the configuration 1 Lag and 100 Epoch. The data counts are given in the table. The results are got with the unit of byte.

#### 4.4.1.6 LSTM Test 6

**Table 4.6: LSTM test 6**

| | Dataset | Time Period | Data Count | Lag | Epoch | RMSE |
|---|---|---|---|---|---|---|
| **1st Day** | 4 | 1-Day | 1440 | 1 | 100 | 16126396.91 |
| **2nd Day** | 4 | 1-Day | 1436 | 1 | 100 | 10887676.78 |
| **3rd Day** | 4 | 1-Day | 1436 | 1 | 100 | - |
| **4th Day** | 4 | 1-Day | 1440 | 1 | 100 | 30435664.06 |
| **5th Day** | 4 | 1-Day | 1440 | 1 | 100 | 61995280.53 |
| **6th Day** | 4 | 1-Day | 1440 | 1 | 100 | 121738469.86 |
| **7th Day** | 4 | 1-Day | 1440 | 1 | 100 | 5120305.95 |
| **Average** | | | | | | **41050632.34** |

*Table 4.6* informs us about a sequence of 1-Day tests which are done with the configuration 1 Lag and 100 Epoch. The data counts are given in the table. The results are got with the unit of byte. 3rd Day result is discarded because this dataset does not work with the ARIMA.

#### 4.4.1.7 LSTM Test 7

**Table 4.7: LSTM test 7**

| | Dataset | Time Period | Data Count | Lag | Epoch | RMSE |
|---|---|---|---|---|---|---|
| **1st Day** | 5 | 1-Day | 1440 | 1 | 100 | 24151364.33 |
| **2nd Day** | 5 | 1-Day | 1436 | 1 | 100 | 91055061.62 |
| **3rd Day** | 5 | 1-Day | 1436 | 1 | 100 | 128756738.28 |
| **4th Day** | 5 | 1-Day | 1440 | 1 | 100 | 37148327.59 |
| **5th Day** | 5 | 1-Day | 1440 | 1 | 100 | 49934391.55 |
| **6th Day** | 5 | 1-Day | 1440 | 1 | 100 | 251217652.98 |
| **7th Day** | 5 | 1-Day | 1440 | 1 | 100 | 14597938.50 |
| **Average** | | | | | | **85265924.97** |

*Table 4.7* informs us about a sequence of 1-Day tests which are done with the configuration 1 Lag and 100 Epoch. The data counts are given in the table. The results are got with the unit of byte.

### 4.4.2 ARIMA Tests

The first two tests are done to determine the best ARIMA configuration. Then, the next five tests are done with all datasets like LSTM. Again, 1-Day time period is used with a sequence of days like LSTM because the longer time period needs the longer computational time as mentioned.

#### 4.4.2.1 ARIMA Test 1

**Table 4.8: ARIMA test 1 configuration**

| **Dataset** | **Dataset 1** |
|---|---|
| **Data Period – Data Count** | 1 Hour – 60 |
| **P, D, Q** | 1, 1, 0 |
| **Test RMSE** | **18083689.41** |

*Table 4.8* gives the certain information about the data and the algorithm configuration. 1-Hour time period data containing 60 time series data is used with the ARIMA configuration P: 1, D: 1 and Q: 0.

**Figure 4.41: ARIMA test 1**

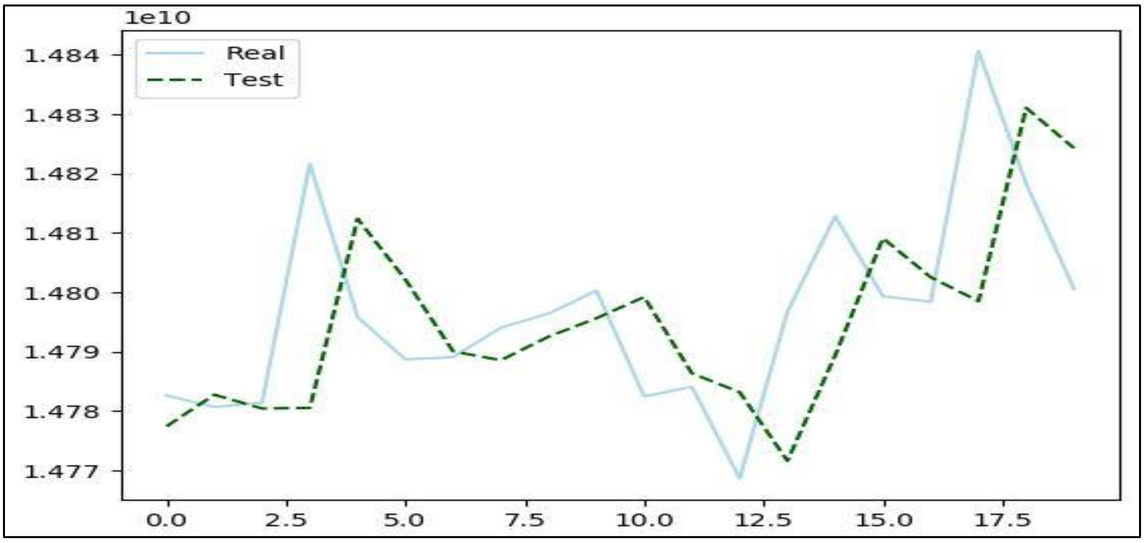


*Figure 4.41* shows the result of the test that is done with the configuration mentioned *Table 4.8*. As seen from the legend, blue colour means the real data and the green one means the test prediction.

#### 4.4.2.2 ARIMA Test 2

**Table 4.9: ARIMA Test 2 configuration**

| **Dataset** | **Dataset 1** |
|---|---|
| **Data Period – Data Count** | 1 Hour – 60 |
| **P, D, Q** | 3, 1, 0 |
| **Test RMSE** | **16331034.25** |

*Table 4.9* gives the certain information about the data and the algorithm configuration. 1-Hour time period data containing 60 time series data is used with the ARIMA configuration P: 3, D: 1 and Q: 0.

**Figure 4.42: ARIMA Test 2**

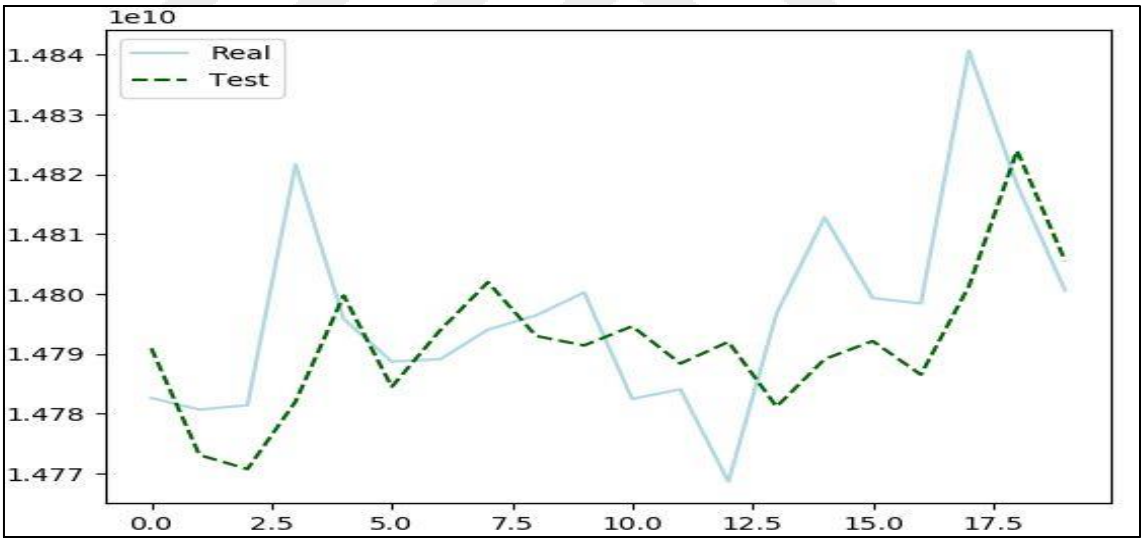


*Figure 4.42* shows the result of the test that is done with the configuration mentioned *Table 4.9*. As seen from the legend, blue colour means the real data and the green one means the test prediction.

#### 4.4.2.3 ARIMA Test 3

**Table 4.10: ARIMA Test 3**

| **Dataset** | **Dataset 1** |
|---|---|
| **Data Period – Data Count** | 1 Hour – 60 |
| **P, D, Q** | 5, 1, 0 |
| **Test RMSE** | **17651244.11** |

*Table 4.10* gives the certain information about the data and the algorithm configuration. 1-Hour time period data containing 60 time series data is used with the ARIMA configuration P: 5, D: 1 and Q: 0.

**Figure 4.43: ARIMA Test 3**

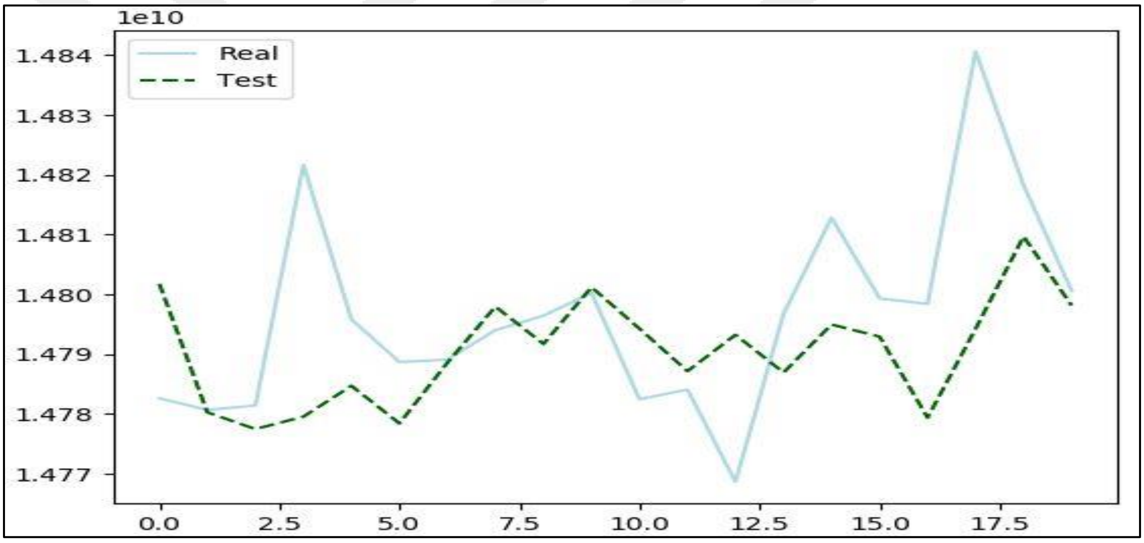


*Figure 4.43* shows the result of the test that is done with the configuration mentioned *Table 4.10*. As seen from the legend, blue colour means the real data and the green one means the test prediction.

### 4.4.2.4 ARIMA Test 4

**Table 4.11: ARIMA Test 4**

| | Dataset | Time Period | Data Count | P, D, Q | RMSE (byte) |
|---|---|---|---|---|---|
| **1st Day** | 1 | 1-Day | 1440 | 3, 1, 0 | 45436295.29 |
| **2nd Day** | 1 | 1-Day | 1440 | 3, 1, 0 | 29089817.32 |
| **3rd Day** | 1 | 1-Day | 1437 | 3, 1, 0 | 93711933.68 |
| **4th Day** | 1 | 1-Day | 1440 | 3, 1, 0 | 24424130.22 |
| **5th Day** | 1 | 1-Day | 1440 | 3, 1, 0 | 32175906.25 |
| **6th Day** | 1 | 1-Day | 1440 | 3, 1, 0 | 202988258.59 |
| **7th Day** | 1 | 1-Day | 1440 | 3, 1, 0 | 55228454.71 |
| **Average** | | | | | **69007828.00** |

*Table 4.11* informs us about a sequence of 1-Day tests which are done with the configuration 3 P, 1 D and 0 Q. The data counts are given in the table. The results are got with the unit of byte.

### 4.4.2.5 ARIMA Test 5

**Table 4.12: ARIMA Test 5**

| | Dataset | Time Period | Data Count | P, D, Q | RMSE (byte) |
|---|---|---|---|---|---|
| **1st Day** | 2 | 1-Day | 1440 | 3, 1, 0 | 7760893.02 |
| **2nd Day** | 2 | 1-Day | 1440 | 3, 1, 0 | 11999031.27 |
| **3rd Day** | 2 | 1-Day | 1436 | 3, 1, 0 | 40591909.30 |
| **4th Day** | 2 | 1-Day | 1440 | 3, 1, 0 | 13661215.74 |
| **5th Day** | 2 | 1-Day | 1440 | 3, 1, 0 | 11069184.00 |
| **6th Day** | 2 | 1-Day | 1440 | 3, 1, 0 | 20127610.27 |
| **7th Day** | 2 | 1-Day | 1440 | 3, 1, 0 | 7661721.89 |
| **Average** | | | | | **16124509.35** |

*Table 4.12* informs us about a sequence of 1-Day tests which are done with the configuration 3 P, 1 D and 0 Q. The data counts are given in the table. The results are got with the unit of byte.

### 4.4.2.6 ARIMA Test 6

**Table 4.13: ARIMA Test 6**

| | **Dataset** | **Time Period** | **Data Count** | **P, D, Q** | **RMSE (byte)** |
|---|---|---|---|---|---|
| **1st Day** | 3 | 1-Day | 1440 | 3, 1, 0 | 348346529.72 |
| **2nd Day** | 3 | 1-Day | 1436 | 3, 1, 0 | 341676876.95 |
| **3rd Day** | 3 | 1-Day | 1436 | 3, 1, 0 | 364095331.43 |
| **4th Day** | 3 | 1-Day | 1440 | 3, 1, 0 | 398491801.21 |
| **5th Day** | 3 | 1-Day | 1440 | 3, 1, 0 | 341263918.43 |
| **6th Day** | 3 | 1-Day | 1440 | 3, 1, 0 | 344697033.57 |
| **7th Day** | 3 | 1-Day | 1440 | 3, 1, 0 | 336440772.60 |
| **Average** | | | | | **353573180.55** |

*Table 4.13* informs us about a sequence of 1-Day tests which are done with the configuration 3 P, 1 D and 0 Q. The data counts are given in the table. The results are got with the unit of byte.

### 4.4.2.7 ARIMA Test 7

**Table 4.14: ARIMA Test 7**

| | Dataset | Time Period | Data Count | P, D, Q | RMSE (byte) |
|---|---|---|---|---|---|
| 1st Day | 4 | 1-Day | 1440 | 3, 1, 0 | 7646714.44 |
| 2nd Day | 4 | 1-Day | 1436 | 3, 1, 0 | 9905055.44 |
| 3rd Day | 4 | 1-Day | 1436 | 3, 1, 0 | - |
| 4th Day | 4 | 1-Day | 1440 | 3, 1, 0 | 10612086.34 |
| 5th Day | 4 | 1-Day | 1440 | 3, 1, 0 | 30186624.48 |
| 6th Day | 4 | 1-Day | 1440 | 3, 1, 0 | 50846967.33 |
| 7th Day | 4 | 1-Day | 1440 | 3, 1, 0 | 4657961.76 |
| Average | | | | | **18975901.63** |

*Table 4.14* informs us about a sequence of 1-Day tests which are done with the configuration 3 P, 1 D and 0 Q. The data counts are given in the table. The results are got with the unit of byte. 3rd Dataset is discarded because it cause problems with the algorithm.

### 4.4.2.8 ARIMA Test 8

**Table 4.15: ARIMA Test 8**

| | Dataset | Time Period | Data Count | P, D, Q | RMSE (byte) |
|---|---|---|---|---|---|
| 1st Day | 5 | 1-Day | 1440 | 3, 1, 0 | 24169475.77 |
| 2nd Day | 5 | 1-Day | 1436 | 3, 1, 0 | 72188019.60 |
| 3rd Day | 5 | 1-Day | 1436 | 3, 1, 0 | 84019240.30 |
| 4th Day | 5 | 1-Day | 1440 | 3, 1, 0 | 35074656.79 |
| 5th Day | 5 | 1-Day | 1440 | 3, 1, 0 | 42404796.06 |
| 6th Day | 5 | 1-Day | 1440 | 3, 1, 0 | 39457308.44 |
| 7th Day | 5 | 1-Day | 1440 | 3, 1, 0 | 14065486.62 |
| Average | | | | | **44482711.94** |

*Table 4.15* informs us about a sequence of 1-Day tests which are done with the configuration 3 P, 1 D and 0 Q. The data counts are given in the table. The results are got with the unit of byte.

### 4.4.3 Evaluation of LSTM and ARIMA

When assessing the results of the experiments, it can be seen from the Table 4.14 that the results are quite the same but ARIMA is better in contrast to common sense in literature.

**Table 4.16: Evaluation of LSTM and ARIMA**

| Averages | LSTM (byte / GB) | ARIMA (byte / GB) |
|---|---|---|
| **Dataset 1** | 169005684.36 | 69007828,00 |
| **Dataset 2** | 24459571.56 | 16124509.35 |
| **Dataset 3** | 358087982.14 | 353573180.55 |
| **Dataset 4** | 41050632.34 | 18975901.63 |
| **Dataset 5** | 85265924.97 | 44482711.94 |
| **Overall Average** | **135573959.07 byte**<br>**0.12 GB** | **100432826.29 byte**<br>**0.09 GB** |

*Table 4.16* informs us about the average results of the algorithms, LSTM and ARIMA. A number of tests are done with each dataset and get their average results. Finally, overall average results are calculated through the average results. It is seen that ARIMA is better. Since the results are based on byte, the overall average has also the GB conversion. By the one dataset does not work with the ARIMA that is excluded.

### 4.4.4 VMware Tests

In horizontal scaling, new Centos 6.5 VM containing 1 CPU and 1 GB ram is created in about 2:18 minutes with all configurations and powering on. In vertical scaling, RAM hot plug gets about 6 six seconds. However, VMware has a very important constraint that it does not allow hot remove. In migration, host change gets about 6 seconds and VM deletion is done in just 2 seconds.

### 4.4.5 OnApp Tests

In horizontal scaling, again new Centos 6.5 VM containing 1 CPU and 1 GB RAM is created at in about 2:45 minutes with all configurations and powering on. In vertical scaling, RAM hot plug or hot remove gets just 1 second, however, there are certain constraints on of which  is that some OSs do not allow hot plug, another of which is that hot plug addition should be maximum 16 times the initial amount. In migration, host change gets 91 seconds and VM deletion is done in about 4 minutes.

### 4.4.6 Evaluation of VMware Hot Plug vs OnApp Hot Resize

Since VMware does not allow hot memory remove, it is not eligible to full dynamic resource management but it can be used in some business plans. OnApp is more suitable platform for that kind of system though it has certain constraints.

# 5. RESULTS AND DISCUSSION

At the beginning of the work we stated that there were some resource management methods such as static management, horizontal scaling, vertical scaling and hybrid scaling and the hybrid one seemed to be the more efficient than the others because of the fact that it uses advantages of the vertical scaling and use the horizontal scaling when it is needed. Also we said that without these types there are problems such as scale amount, scale timing and scale duration. As a solution, we have proposed the predictive hybrid scaling.

From the Section 4.2.4, Dataset 4 is very clear reason of these issues mentioned above. Even in one day period, it has about 45 GB available memory which is an exact resource wastage. This is the answer of why we should not manage the resources statically. Also, from the *Figure 4.31*, we see sharp increases about usage level from about 2.03 GB to 61.38 GB, which is the reason of why we should not use any kind of reactive solution because it may probably not be effective. This may also the reason of why we should prefer predictive hybrid scaling than predictive vertical scaling. When resource increases are above the limits of the PM, horizontal scaling will be required.

*Table 4.16* shows the results of the experiments about prediction algorithms. Both of LSTM and ARIMA have promising results except that ARIMA is better. From the results we see that the average results of the LSTM tests is the 135573959.07 byte which is the 0.12 GB and the average results of the ARIMA tests is the 100432826.29 byte which is the 0.09 GB. Thus, both of them can be used in real life situations. Also from the Section 4.4.4 and Section 4.4.5 we can see the scaling performances of the VMware and OnApp orderly. Although both VMware and OnApp have some restrictions, process times of them are very good. However, VMware is able to hot memory remove that may be resolved in the future. Moreover, since we know the scale duration based on the cloud platform, the scale time cane be selected safely according to the related business rules.

# 6. CONCLUSION

This work shows why resources should not be managed statically, why a scaling type is required, why hybrid is better than vertical or horizontal individually and why scaling should be predictive rather than reactive. In the experiments, five real datasets got from Radore Data Center are used with the prediction algorithms LSTM and ARIMA. Although both of them have good results ARIMA’s score is better. However, one dataset does not work with the ARIMA that is discarded from the results. VMware and OnApp are used as cloud infrastructures. Again both of them good results but VMware does not support hot remove on memory that is a problem which may be solved in the future. OnApp has better performances than VMware. As a result, the system did its job successfully.

# REFERENCES

### *Books*

Springer, 2010. Virtualization. In: B. Furht & A. J. Escalante, eds. *Handbook of Cloud*. New York: Springer Science+Business Media, LLC, p. 9.

### ***Periodicals***

Calheiros, R. N., Masoumi, E., Ranjan, R. & Buyya, R., 2014. Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS. *IEEE Transactions on Cloud Computing,* 3(4), pp. 449 - 458.

Chenhao, Q., Calheiros, R. N. & Buyya, R., 2017. Auto-scaling Web Applications in Clouds: A Taxonomy and Survey. *Auto-scaling Web Applications in Clouds: A Taxonomy and Survey,* 14 September, p. 1.

Hochreiter, S. & Schmidhuber, J., 1997. Long Short - Term Memory. *Neural Computation,* 9(8), pp. 1735-1780.

Lu, Y., Panneerselvam, J., Liu, L. & Wu, Y., 2016. RVLBPNN: A Workload Forecasting Model for. *Scientific Programming,* 2016(5635673), p. 1.

Vazquez, C., Krishman, R. & John, E., 2015. Time Series Forecasting of Cloud Data Center Workloads for Dynamic Resource Provisioning. *Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications,* 6(3), pp. 87-110.

Xiao, Z., Song, W. & Chen, Q., 2013. Dynamic Resource Allocation Using Virtual. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,* June, 24(6), p. 1107.

Zhang, Y., Xiong, R. & He, H., 2018. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE,* 0(0), pp. 1-1.

### *Other Sources*

Azure, M., 2018. *What is cloud computing.* [Online] Available at: https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/ [accessed on 15 04 2018].

Bontempi, G., 2013. *Machine Learning Strategies for Time.* [Online] Available at: http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf [accessed on 16 04 2018].

Brownlee, J., 2017. *How to Create an ARIMA Model for Time Series Forecasting with Python.* [Online] Available at: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/ [accessed on  16 04 2018].

Butler, B., 2013. *NetworkWorld.* [Online] Available at: https://www.networkworld.com/article/2163667/cloud-computing/onapp--the-most-popular-cloud-platform-you-ve-probably-never-heard-of.html [accessed on 16 04 2018].

Dalinina, R., 2013. *Introduction to Forecasting with ARIMA in R.* [Online] Available at: https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-data-science-tutorials [accessed on 17 04 2018].

DigitalSingleMarket, 2014. *Cloud Service Level Agreement.* [Online] Available at: http://ec.europa.eu/newsroom/dae/document.cfm?action=display&doc_id=6138 [accessed on 16 04 2018].

DL4J, 2017. *A Beginner’s Guide to Recurrent Networks and LSTMs.* [Online] Available at: https://deeplearning4j.org/lstm.html [accessed on 17 04 2018].

DL4J, 2017. *Introduction to Deep Neural Networks (Deep Learning).* [Online] Available at: https://deeplearning4j.org/neuralnet-overview.html [accessed on 17 04 2018].

Erradi, A. & Kholidy, H. A., 2017. *An efficient hybrid prediction approach for predicting cloud consumer resource needs.* Agadir, Morocco, IEEE.

Gupta, P., Samvatsar, M. & Singh, U., 2017. *Cloud Computing Through Dynamic.* Coimbatore, India, IEEE.

Hostbill, 2018. *Onapp Cloud.* [Online] Available at: https://hostbillapp.com/feature/onapp/ [accessed on 16 04 2018].

Huerta, G., 2006. *tseries.* [Online] Available at: http://www.math.unm.edu/~ghuerta/tseries/week4_1.pdf [accessed on 16 04 2018].

IBM, 2018. *IBM PowerVM.* [Online] Available at: https://www.ibm.com/us-en/marketplace/ibm-powervm [accessed on 16 04 2018].

IBM, 2018. *IBM z/VM.* [Online] Available at: https://www.ibm.com/it-infrastructure/z/zvm [accessed on 16 04 2018].

IBM, 2018. *Z/VM Overwiev.* [Online] Available at: http://www.vm.ibm.com/overview/ [accessed on 16 04 2018].

İsmail, H. A. & Riasetiawan, M., 2016. *CPU and Memory Performance Analysis on Dynamic and Dedicated Resource Allocation using XenServer in Data Center Environment.* Yogyakarta, Indonesia, IEEE.

Kvm, 2018. *Kvm.* [Online] Available at: https://www.linux-kvm.org/page/Main_Page [accessed on 16 04 2018].

Matthias, M., Klink, M., Tomforde, S. & Hahner, J., 2016. *Predictive Load Balancing in Cloud Computing.* Wurzburg, Germany, IEEE.

Mell, P. & Grance, T., 2009. *The NIST Definition of Cloud Computing.* [Online] Available at: https://www.nist.gov/sites/default/files/documents/itl/cloud/cloud-def-v15.pdf [accessed on 15 04 2018].

Olah, C., 2015. *Understanding LSTM Networks.* [Online] Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [accessed on 17 04 2018].

Onapp, 2018. *THE COMPLETE CLOUD MANAGEMENT PLATFORM.* [Online] Available at: https://onapp.com/cloud-management-platform/#onapp-cloud [accessed on 16 04 2018].

PSU, 2018. *Autoregressive Models.* [Online] Available at: https://onlinecourses.science.psu.edu/stat501/node/358 [accessed on 16 04 2018].

Pushkar, S. & Dubno, M., 2018. *OnApp 5.0 API Guide.* [Online] Available at: https://docs.onapp.com/display/50API/OnApp+5.0+API+Guide [accessed on 26 04 2018].

Ranger, S., 2018. *Zdnet.* [Online] Available at: https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-from-public-and-private-cloud-to-software-as-a/ [accessed on 15 04 2018].

Rouse, M., 2013. *searchitoperations.* [Online] Available at: https://searchitoperations.techtarget.com/definition/Zabbix [accessed on 16 04 2018].

Rouse, M., 2017. *SearchCloudComputing.* [Online] Available at: https://searchcloudcomputing.techtarget.com/definition/cloud-computing [accessed on 15 04 2018].

Shaikh, G. E. & Shrawankar, U., 2015. *Dynamic memory allocation technique for virtual machines.* Coimbatore, India, IEEE.

Sommer, M., Klink, M., Tomforde, S. & Hahner, J., 2016. *Predictive Load Balancing in Cloud Computing Environments Based on Ensemble Forecasting.* Wurzburg, Germany, IEEE.

Stamford, C., 2017. *Gartner.* [Online] Available at: https://www.gartner.com/newsroom/id/3815165 [accessed on Saturday April 2018].

Stergiou, C. & Siganos, D., 1989. *NEURAL NETWORKS.* [Online] Available at: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html [accessed on 21 04 2018].

Techopedia, 2018. *techopedia.* [Online] Available at: https://www.techopedia.com/definition/26951/pay-as-you-go-payg [accessed on Saturday April 2018].

techopedia, 2018. *What is VMware.* [Online] Available at: https://www.techopedia.com/definition/16053/vmware [accessed on 16 04 2018].

Tholeti, B. P., 2011. *Learn about hypervisors, system virtualization, and how it works in a cloud environment.* [Online] Available at: https://www.ibm.com/developerworks/cloud/library/cl-hypervisorcompare/ [accessed on 16 04 2018].

Thomas, A., 2017. *Adventures in Machine Learning.* [Online] Available at: http://adventuresinmachinelearning.com/neural-networks-tutorial/ [accessed on 21 04 2018].

Toque, F., Come, E., El Mahrsi, M. K. & Oukhellou, L., 2016. *Forecasting Dynamic Public Transport Origin-Destination Matrices.* Rio de Janeiro, Brazil, IEEE.

VMware, 2018. *ESXi.* [Online] Available at: https://www.vmware.com/products/esxi-and-esx.html [accessed on 16 04 2018].

VMware, 2018. *Getting Started with vCenter Server and the vSphere Web Client.* [Online] Available at: https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.solutions.doc%2FGUID-4DB6A316-4A70-49A1-926A-851C5F160378.html [accessed on 16 04 2018].

Wu, J. & Sun, S. L., 2017. *A Dynamic Memory Allocation Approach for Virtualization Platforms.* Beijing, China, IEEE.

XenProject, 2018. *Xen Project.* [Online] Available at: http://www-archive.xenproject.org/products/xenhyp.html [accessed on 16 04 2018].

Zabbix, 2017. *Zabbix Documentation 3.4.* [Online] Available at: https://www.zabbix.com/documentation/3.4/manual/introduction/whatsnew340#new_dashboards [accessed on 16 04 2018].

Zabbix, 2018. *Zabbix Documentation.* [Online] Available at: https://www.zabbix.com/documentation/3.0/ [accessed on 23 04 2018].

Zhang, S. et al., 2017. *Selling Reserved Instances through Pay-as-you-go.* Honolulu, HI, USA, IEEE.

# CURRICULUM VITAE

**Name & Surname**: Fatih KÜÇÜKKARA

**Permanent Address**: Uğur Mumcu Neighborhood, 2117 Street, Number: 5, Apartment: 6, Sultangazi / İstanbul

**Place and Year of Birth**: Bornova / İzmir

**Foreign Language**: English (Advanced), Japanese (Beginner)

**Primary Education**: Mediha Mahmutbey Primary School, 2002

**Secondary Education**: Konak Atatürk Anatolian Trade Vocational High School, 2006

**Undergraguate**: Boğaziçi University (Computer Education and Educational Department), 2013

**Name of Institute**: Institute of Science

**Name of Master’s Program**: Computer Engineering

**Working Life** :

Radore Data Center (Software Development Team Lead) - 2016 Ağustos – Continue

Çiçek Sepeti (Software Development Specialist) - 2015 Eylül – 2016 Temmuz

Belbim (Senior Software Developer) - 2013 Temmuz – 2015 Ağustos

SK Teknoloji (Software Developer) - 2013 Şubat – 2013 Temmuz

Ritma Teknoloji (Software Developer) - 2012 Eylül – 2013 Şubat