

# Predictive Resource Allocation Scheme with Pattern Matching

Abdulhamza A. Abdulkarim

Al-Mamoon University College, Department of Computer Engineering Techniques

*Dr.hamza98@yahoo.co.uk*

## Article Info

Volume 83

Page Number: 3802 - 3816

Publication Issue:

March - April 2020

## Abstract:

Cloud provides huge processing power, immense memory and great storage for its customers on demand but a Cloud setup never have enough amount of resources actually. To accomplish the basic rule of Cloud game, always provide resources as customer required, Cloud play tricky games. Cloud system, keep eye on resource usage and maintain a fragile balance among supply and demand. The availability of resources, elasticity and quality of services increase the profit and monetary benefits of Cloud owner. The following treatise presented a simple but power technique that enables a Cloud system setup its balance of supply and demand with less extra efforts. This is a simple pattern matching method that provide the prediction of resource usage over a time period. In this method, system setup groups of resources and then deploy these resources initially. Later the usage pattern extracted from each group and prediction of next utilization of this bundle on the base of KNN classifier. The output was perpetual in its nature because system have no initial information to predict but every course of execution it increased its ability to get closer of actual requirements. The final results showed the maximum up to 98.5% and reduce the latency as well.

**Keywords:** Matching, Pattern, Predictive Resource.

## Article History

Article Received: 24 July 2019

Revised: 12 September 2019

Accepted: 15 February 2020

Publication: 23 March 2020

## I INTRODUCTION

Cloud system consist on physical machines (PMs) that host virtual machines (VMs) [1], some cloud systems use containers [2] for such purpose but the overall results remain same in broader contexts. Each VM is a set of physical resources that provided to customer on demand. Normally each customer has its own VM that work like real machine. If customer required more physical resources then the configurations of VM can change and on the fly [3].

In fact, a physical machine host multiple VMs or containers simultaneously. There are multiple benefits of these VMs is the robustness. These machines can be changed their configuration, they can migrate to some other physical machine, a single virtual machine can be executed on multiple

physical machines and these physical machines are even geographically dispersed [4].

Cloud have multiple models to implement including conventional datacenter where servers are employed, or many clouds can share their resources according to requirements and formed a Federated Cloud [5]. Another form of distributed computing provide aid to Cloud for assistances named Edge computing [6] and smart devices also integrate each other in Cloud method [7].

Each VM initially setup as pool of configurable resources like physical machine. It has CPU cores, it has memory, it has storage, it might contain the GPU share, it looks like an independent network node [8]. In similar way it contains its own

operating system and other software components including servers, mail clients and applications that interface of usage for a customer [9] [10].

The obvious target of research work is, provide a method that analyze and predict the resource requirement from cloud and offer these set of resources to user. The proposed system creates bundles of resources like VM and whenever a user demand resource, the system offers the package according to prediction.

## II Dimensions of Virtual Machine Based Solutions

VM is one of the solutions of problem in Cloud resource management because, the initial configuration of VM always lesser

then its host physical machine and later on demand, it can be modified and updated and event its resources can be greater than host machines by integrating the virtual resources. There are multiple aspects and dimension of the study and application of this solution as described following.

### 1.1.1. Virtual Machines VS Containers VS Para Virtualization

In Cloud system, there heterogenous users produce heterogenous workload with multiple configuration of resources. To cop this issue, three main solutions were proposed [11] [12]. First is Virtual Machine (VM), that an abstract machine to user without no underlying software and hardware configuration of physical machines

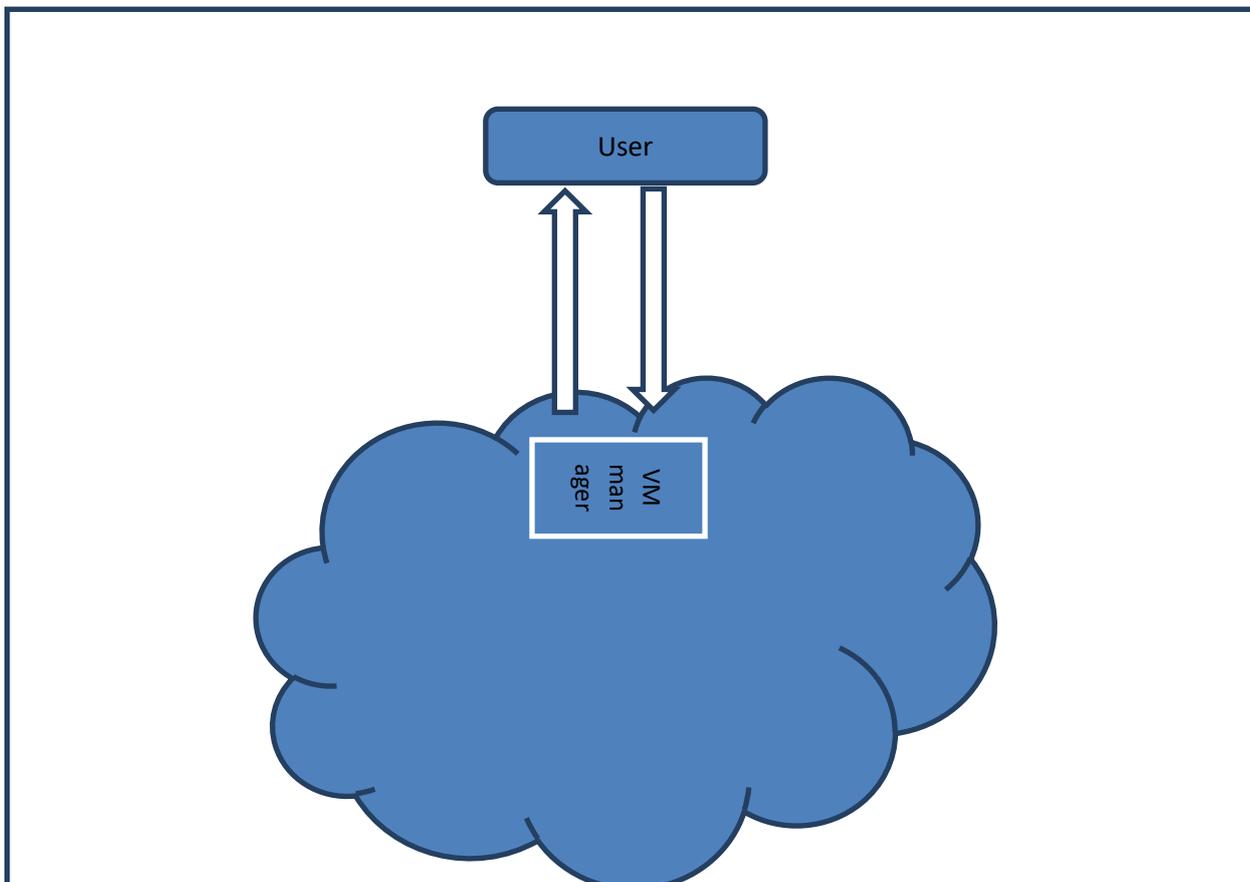


Figure 1. VM Machines, an Overview

An operating system of VM may be differ from its host machine and even a VM can host some other VMs with more different setups [13] [14] [15]. These virtual machines can communicate each other and enable to work like client and server model.

On the other hand, Containers work on abstract operating system. No separate type of platform is provided by host. The application fragmentation is provided by virtual memory. Container's

approach is used by larger vendors and in such cloud systems that required less variate workload [16] [17] [18]. Dockers [19], Ansible [20] and Hyper-V [21] are some solutions provided as Containers.

The third method is mixed approach of VM and Containers. In fact, this is applying Container approach over a VM. It is fully virtualized it will not run as complete isolated environment and use lesser resources than a complete system.

Table 1 List of popular Containers [22]

Name	Developer	Features
Dockers	Dockers	Infrastructure, Orchestration, Portable
AWS Fargate	Amazon	Scalable, Multiple starts, Heterogeneity
Kuberneness	Google	Hybrid, Docker support, GPU support
ECS	Amazon	Windows support, Auto recovery
LXC	Canonical Ltd	Linux based, high density

CoreOS

Red Hat

Gento Linux, Chrome OS

### 1.1.2. Virtual Machine Migration

Virtual machines can be transferred from one physical node to some other node. Multiple factors involved in such operation such as load balancing [23], over and under provisioning [24], elasticity [25], cost and service break downs [26]. Migration is really handy in various cases but system must be careful about migrations and complete analysis because next hosts might not be ready for such change. They might be overloaded already; they are geographically far and more bandwidth consumption cost added to task and latency issues [27].

### 1.1.3. Update Configuration

User requirements can be change on runtime, they required more or less resources. A good cloud manger takes care of these instantaneous changes and update the assigned resources to VM.

Updating configuration also enables pay as you go model where user pay the resource utilization instead of static rent of Cloud. A predictive approach was introduced by Murthy et al [28]. With help of workload generation toolkit, RAIN and Cloud benchmark application Olio they produced results at 90-92 percent accuracy. A layered approach was presented by Armstrong et al [29] that provide contextual changes in VM setup. Almost all VM tools including VMWare also provide such interface that enables a Virtual machine to changes its management over the fly.

### 1.2. K-Nearest Neighbor Classifier

K-Nearest Neighbor (KNN) is one of the fundamental classifiers that work as supervised learning method [30] [31]. The classifier, first creates classes of given training dataset and when a testing dataset supplies, it calculates the candidate value of given instance. In next step it

calculates the distances with  $K$  number of training set instances and find which class instances are selected more. Ultimately, the distance calculating

technique and selecting  $K$  is crucial in all the working of classifier [32].

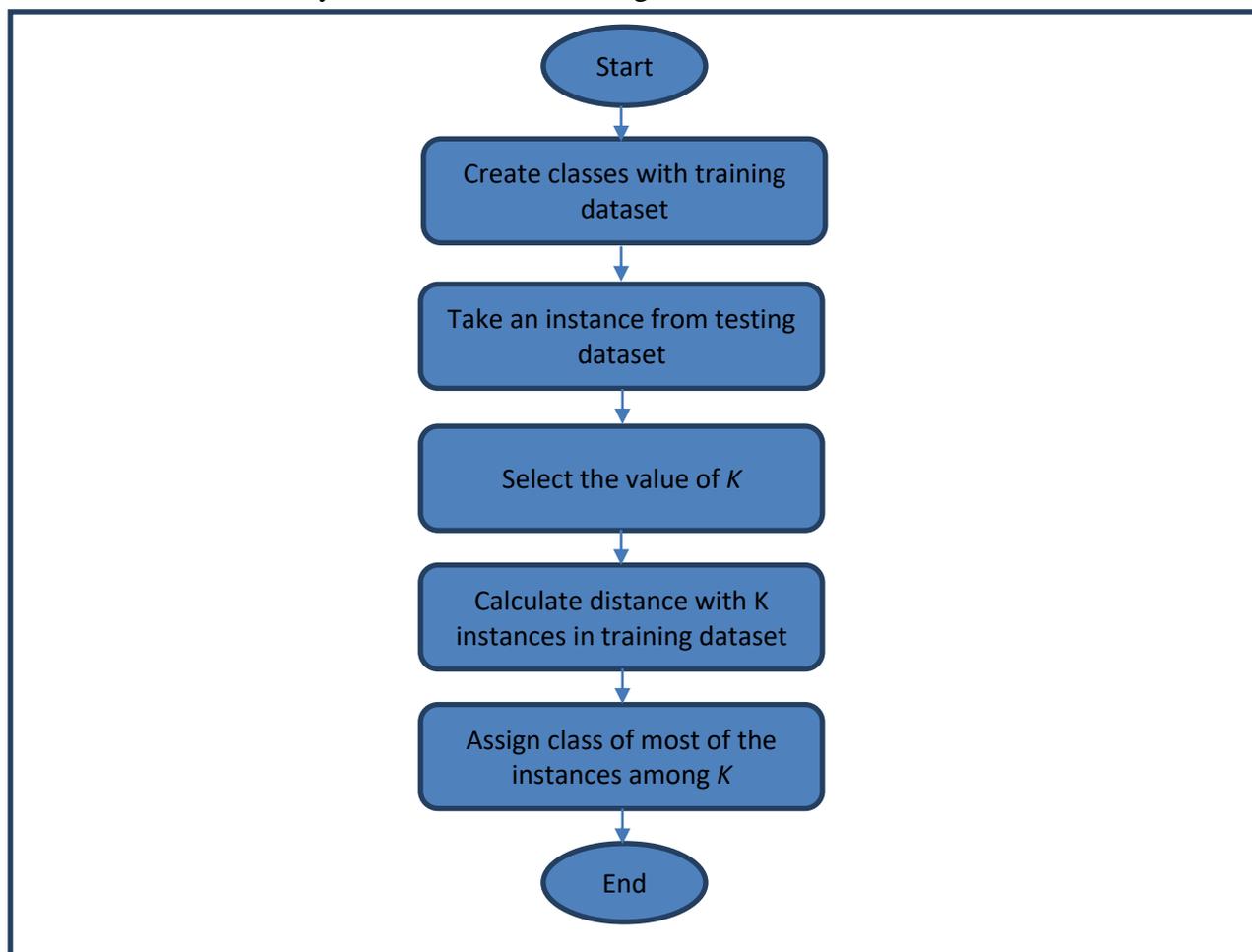


Figure 2. K-NN Classifier

The distances can be calculated by multiple methods. We Euclidian method.

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 \dots}$$

$x_i$ : Instance of Testing dataset

$y_i$ : Instances of Training dataset

$i$ : Feature of the classes

There might be  $n$  number of features might be selected from class. There might be some insignificant features available in dataset then these can be ignored. Similarly, the value of  $K$  also effects the results. The best of  $K$  is total

instances in training set for best accuracy but this can be affected the performance and running time. To get best comparison among the results, we can use different values of  $K$  for different evaluations.

KNN classifier have multiple variants that work with supervised, unsupervised and regression classification. Similarly the value of  $K$  can be set institutively without using any special mechanism.

### 1.3. CloudSim

CloudSim is an open source simulator developed in Java. In this simulator we can create VMs, cloud centers and workload and apply that

workload on these virtual machines and find the performance of virtually created instances. CloudSim provide us some facility to provide our own job scheduling method to implement and then we can evaluate the effectiveness of proposed method [33].

To test our framework on CloudSim we created our workload called *Cloudlets*. Basically, *Cloudlet* class have four constructors with different parameters.

To run our simulation and evaluation, we use following parameters [34].

1. *Cloudlet ID*: Auto generated integer starts from 1
2. *Cloudlet Length*: Number of instructions in job
3. *Pes Number*: Number Processors required to execute this job
4. *Cloudlet File Size*: Size of Data File. This will to find the consumption of storage and bandwidth
5. *User ID*: This is broker id, a human or machine that create a cloud and workload
6. *Cloudlet Output Size*: The output of Cloud workload
7. *Status*: This field indicates the situation of a Cloudlet. When some workload applied to machine, machine can produce different reactions. List is following
  - **CREATED**: Cloudlet just created
  - **READY**: Machines is ready to start executing workload
  - **QUEUED**: Cloudlets are added to queue
  - **INEXEC**: Cloudlet is executing

- **SUCCESS**: Cloudlet successfully executed. This is status is important for our system
- **FAILED**: Cloudlet could not be completed due to any reason
- **CANCELED**: Job differed by user
- **PAUSED**: Job halted for a while
- **RESUMED**: Cloudlet resumed after paused
- **FAILED\_RESOURCES\_UNAVAILABLE**: This is also important for our system. We will mark our offered package as accepted or rejected

8. *Execution Start & Finish Time*: Time stamp for starting and Ending of a Cloudlet
9. *Record*: If cloudlet is executed at different resources then either we want to record its parameters or not
10. *Index*: ID of currently using resource
11. *VM ID*: Virtual Machine ID created for task
12. *Bandwidth Cost*: Calculate bandwidth costing using by Cloudlet. Normally File Size will be use for this purpose

### III BACKGROUND

There a rich set of research work, studies, and proposed methodologies are available for resource prediction. Some of the recent and significant works are presented here.

Sood&Sandu developed a Matrix based proactive resource provisioning system for Mobile Cloud Computing [35]. They generate the requirement matrix for each user and calculate the prediction by using Artificial Neural Network

(ANN).

Bayesian model based solution provided by S Kumaraswamy & Mydhili K Nair [36] that predicts the VM requirements for application. The system used the CPU and memory as bench mark and try to collect the data samples from all datacenters of a Cloud in different sessions of time that span days to weeks.

In [23] Chen & Wang presented a hybrid method for short-term host utilization. The method works on Ensembled Empirical Mode Decomposition (EEMD), Run Test (RT) & ARIMA (AutoRegressive Integrated Moving Average). The method was evaluated on accuracy, effectiveness and time-cost analysis.

Deep learning also providing solutions in Cloud arrangements. The prediction mechanism was developed by Feng Qiu [37] using Deep Believe Network (DBN) with multi layered Restricted Boltzman Machine (RBM) and regression layer.

“Virtual Machines Placement Using Predicted Utilization of Physical Machine in Cloud Datacenter” was offered by Varun Barthwal et al [38]. They calculate the workload over a physical machine and generate predictions of the base of this log.

Power consumption was also an important part of virtual machine placements. This problem was addressed by Veni&Bhanu with Non-Linear Support Regression [39]. They compare their method with Linear Support Regression and showed better performance.

Consolidation based method was proposed by FahimehFarahnakian et al [40] that implemented with regression based approach what was testified with heuristic and meta heuristic approaches.

A novel approach using ARIMA model was introduced by Alzamil&Djemame [41] showed better performance in energy predictions. This

technique was use patterns to find future parameters and get MPE up to 2.58.

Although the virtual machines have no physical properties like physical machines but, VMs put some physical impacts on environment as well. Wu et al [42] tried to ascertain thermal foot prints in Cloud datacenters by VMs. LIBSVM was integrated with Radial Basis Function (RBF) and compute the predictions of temperature by physical machine.

An experiment over Google trances was conducted and studied by Dabbagh et al [43]. There were two objectives were under considerations by authors. First, reducing the workload on physical machines by monitoring the virtual machines and second was migrating the virtual machines efficiently and minimizing the active number physical machines by using proper prediction.

Virtual Machine consolidation problem was addressed by FahimehFarahnakian [44] that solved issues in Green Cloud computing. This KNN regression-based model that use problem as bin-packing.

Tarek Mahdhi&HaithemMezni [45] target IaaS and develop a model using KDE for forecasting of future requirements. They also target apply their method over VM and AKKA toolkit and obtained some significant results.

A split technique was exhibited by [46] for mobile devices. The aim was, enabling larger workload entertainment on smart objects. For this purpose, application was written in client/server model where heavy operations perform on server and lighter processing was held by the device itself.

Chen Z et al [47] introduced fuzzy neural network method that used fuzzy c-mean. Subtractive clustering was collaborated with FNN and produced better future predictions.

Power issues were addressed by Younge et al [48]. The solution was simply trying to find the already running nodes. Then assign workload these nodes for maximizing the consumption of their resources.

Profit maximization was targeted in [49] with the help of integrating multiple techniques. This approach uses dynamic and static methods. Dynamic method defined at run time when client demands resources while static approach work on reservations of resources in advance. Stochastic Integer Programming was foundation while SAA (Sample Average Approximation), Hungarian method, Benders Decomposition and Deterministic Equivalent were deployed.

Energy optimization covered in [50] by Cuervo et al for mobile phones. The technique named MAIU (). This was also a split model that work both client and server. The application identified the workload and shift the heavy workload on server and remaining part execute on device.

#### IV PROPOSED FRAMEWORK

Table 2 Sample Packages

Package ID	Resources	Description
P1	CPU core: 4 RAM: 1GB Storage: 16GB Bandwidth: 2MB	This is huge resource with some larger volumes
P2	CPU core: 2 RAM: 512 MB Storage: 8GB Bandwidth: 1MB	Medium size pool
P3	CPU core: 1 RAM: 256MB Storage: 4GB Bandwidth: 512 KB	Small size pool
...	...	..

It is not necessary that each package have only one of these sets. It might be change under the light of record and logs. System have multiple set of package configurations in its knowledge base

Proposed framework is simple in its nature that logged the resource usage and offer the set of resources to customers for their requirements. If customer applications consider it best available package then system count it as success and if it is not that system marked it as fail. Consequently, the operations flows and pattern generated of success and failure period of time, the system automatedly learn which resources are best fit for which kind of workload.

#### 1.4. Packages

Package is set of selected resources that contain in the form of VM or container. Each package contains four components including CPU core(s), memory, storage and Bandwidth. System further define different set of packages with respect to volume. Some packages are really huge in size and setup with larger resources while some resources are small in containments. Here is the list of some sample packages.

##### 1.4.1. Sample Package Set

and will employed on requested time.

##### 1.4.2. Log Data

A system component is deployed to keep the

record of logged data like a database. This data contains the job configuration, offered package configuration and response of offer. This data continuously logged into database and further use future. In fact, this database only starts of setup just after system deployment. At this point not previous record is available then system will use default configurations.

### 1.4.3. Datasets

For development purposes, the datasets stored in XML file that was read and write by program. These XML files work like database that included predefined packages, generated cloudlets and results of proposed packages to cloudlet. XML are simple text files and normally, XML can be read and write all programming languages including Java.

```
<packages>
<package id="P1" default = "default">
  <cores>4<cors>
  <memory>4</memory>
  <storage>16</storage>
  <bw>2</bw>
  </package>
</packages>
```

Figure 3.package.xml

```
<offers>
  <offer id="1">
    <cloudlet>
      <id> 1 </id>
      <cpu> 1 </cpu>
      <length> 1.2 </length>
      <filesize> 1.2 </filesize>
    </cloudlet>
    <package id = "1" />
  <response> true </response>
</offer>
</offers>
```

Figure 4. Offers to Cloudlets

In offer.xml, we stored cloudlet with its ID, CPU required, number of instructions included in Cloudlet (job) and file size. After running each cloudlet, system will record its response “true“ or “false”. If cloudlet accepted then it will record as true other wise false.

“package.xml” will be create at start of experiment once and use during evaluation while offer.xml will generate during the execution of program and continuesly updated and read to develop the learning model.

### 1.5. System Components

Overall system architecture is elaborated in Fig x. The core component is Package Manager (PM) that generates and offers the packages. For the first time when no historical data is available for analysis, PM create resources pools intuitively. All the configurations are predefined and whenever system demands some virtual machine the predefined configuration will be offered. Application can accept or defer the offer. In both cases monitor mark the result and save it. Every

now and then, Package manager offer resources and register the results and each time system will offer the resources on the base previous entries. System will show some initial results as incorrect but as well as its knowledge base increased, the prediction frequency also approaches to better levels.

### 1.5.1. Configuration Manager

Configuration generates the configurations for next job. These configurations are prepared after the resolution from Monitor. If no previous data is available then configuration starts from default configuration. Configuration manager prepare these settings by KNN algorithm that take responses, accepted or rejected, and match this

pattern with requirements.

### 1.5.2. Monitor

Monitor component generates the log of accepted and rejected packages and also record the requirements of job as well. This record will help the configuration manager. Monitor continuously keep record of requirements from the jobs in current scenario.

### 1.5.3. Provision Manager

Provision manager offer resources to jobs after configuration manager suggests the resources by configuration. Provision manager just only move, when some jobs demand resources.

## 1.6. Algorithm

```
1. FindBestPackage (Inputs: cloudlet):
2. Components: selected_package, historical_data, label
3. historical_data = GetHistoricalData ()
4. If (historical_data is NULL):
5. selected_package = SelectDefaultPackage()
6. Else:
7. label = Classify (cloudlet)
8. selected_package = SelectPackage(label)
9. Return selected_package

1. Classify (Inputs: cloudlet):
2. Components: Training dataset, K, distances [distance][class]
3. K = GetK()
4. Distances = GetDistancesWithClass(K)
5. Label = GetMostLabeledClasses()
6. Return label

1. SelectPackage (Inputs:label)
2. Foreach (Package Packages)
3. If (package matches label)
4. Return package
```

Fig. 5. Proposed Algorithm

First, the broker will create Cloudlet and send it to Cloud data center. This cloudlet contains number instructions, CPUs required and file size. Cloud data center will invoke classifier and classifier collects data from packages.xml and

offer.xml and build the class for given cloudlet. After calculating the distances and assigning the most suitable class, cloud data center will offer the package to cloudlet. If cloudlet returns the status failed due to unavailable resources then it will be

marked as “false” in offers.xml and if cloudlet executed successfully then it will be marked as true in file. For the next time classifier use this data to again build the prediction and as system runs, the patterns saved in offers.xml and every now and then classifier will increase its accuracy.

In fact, the classifier also written in cloud data center code and it is part of system but it will show separately because it performs its operations independent of the working from normal cloudlet execution and process.

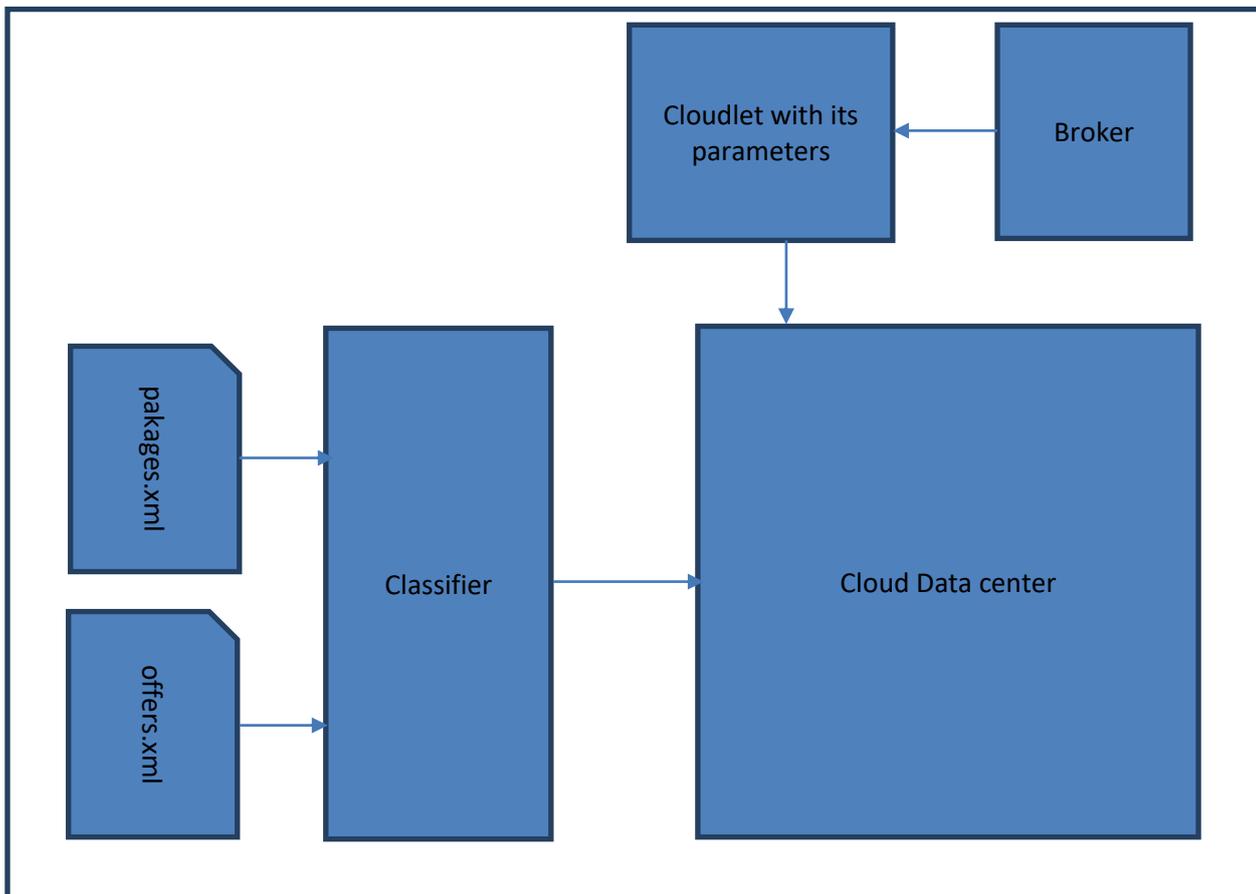


Fig. 6. System Architecture

## V RESULTS

System was tested and evaluated on CloudSim, an open source tools for testing and simulation for Cloud management. The application code written

Java and classification algorithm was also implemented with Euclidian distance. The system largely dependent of past data so it showed some early errors but gradually, results comes better and better. Following simulation setup was used.

Table 3 List of Default Configurations

Package ID	Resources
P1	CPU Cores: 4, RAM: 4 GB, Storage 16 GB, BW: 2MB
P2	CPU Cores: 4, RAM: 2 GB, Storage 8 GB, BW: 1 MB
P3	CPU Cores: 4, RAM: 1 GB, Storage 4 GB: BW: 1 MB
P4	CPU Cores: 2, RAM: 4GB, Storage 16 GB, BW: 2 MB

P5	CPU Cores: 2, RAM: 2 GB, Storage 8 GB, BW, 1 MB
P6	CPU Cores: 2, RAM: 1GB, Storage 4 GB, BW: 512 KB
P7	CPU Cores: 1, RAM: 2 GB, Storage 8 GB, BW: 2 MB
P8	CPU Cores: 1, RAM 1 GB, Storage 4 GB, BW: 1 MB
P9	CPU Cores: 1, RAM 1 GB, Storage 2 GB, BW: 512KB
P10	CPU Cores: 1, RAM: 512 MB, Storage 2GB, WB: 512 KB

The above listed configurations are provided by default when no data is available for lookup. Simulation was executed with different kind and setup of experimental workloads and try to find

the behavior of system. The experiments repeat multiple times and every time, there is boost in accuracy and patterns generated by inputs matches the set of proposed configurations.

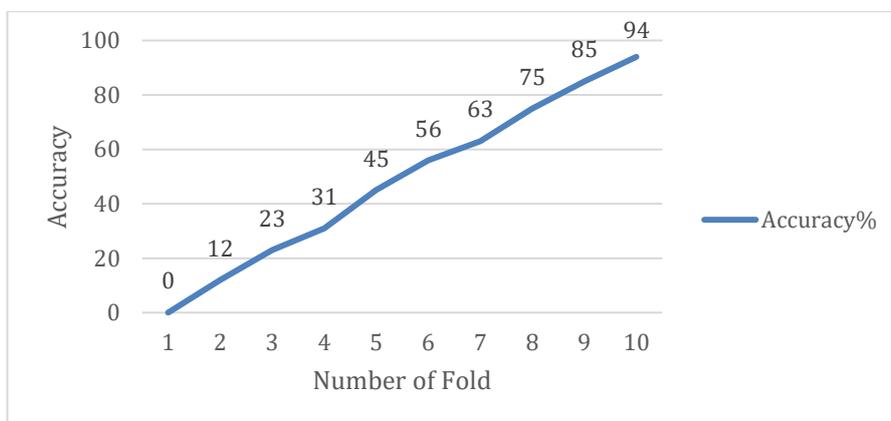


Fig. 7. Accuracy in first 10 Runs

This graph clearly indicates that each time accuracy increased with almost a uniform manner.

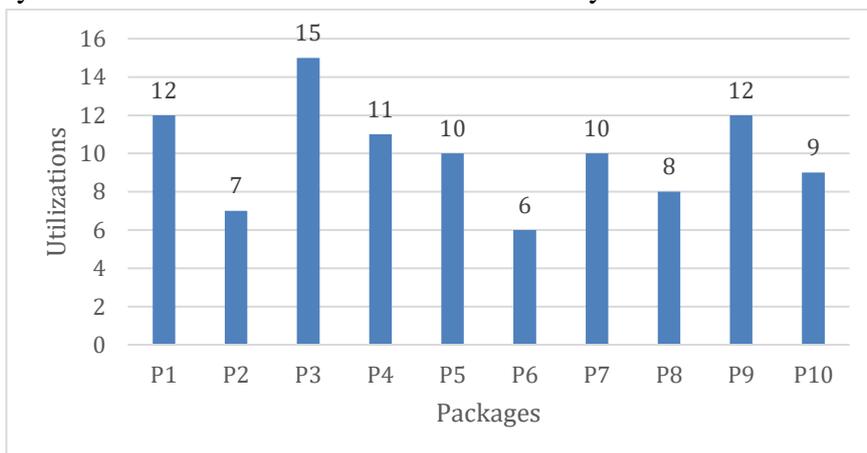


Fig. 8. Resource utilizations in 100 jobs

This graph shows the randomness of package utilization for first 100 jobs. If we combine the graph shown in Fig 4 with this then it is very clear

that utilizations of different packages did not affected accuracy.

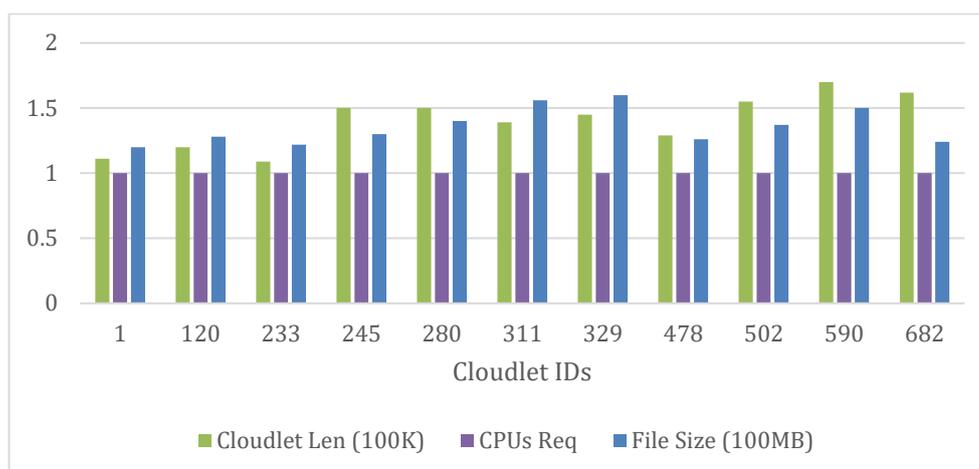


Fig. 9. Some cloudlets with smaller requirements

Here are some cloudlets with minimal parameters. These cloudlets use round about 100 kilo instructions, one CPU and file size just more than 100 megabytes. There are multiple groups of resources can be produced for experimental purposes.

There are some packages which were mostly utilized and some are rarely applied. The most utilized package is P3 (Fig 5) and most unutilized package was P6.

## VI CONCLUSION

The system was designed and evaluated on multiple jobs and try to find the accuracy with various parameters including number of instructions in cloudlet, CPU usage and supplied file size as workload input. There are many limits were also included in the system approach and experiments. Number of features were selected were limited and these can be exceeded. The simulator was used instead of real-world cloud nodes. The system should be examined over real testing environment. There are also multiple options available for simulator as well and system can be tested on multiple virtual environment and find results.

## VII REFERENCES

- [1]. Z. Usmani and S. Singh, "A Survey of Virtual Machine Placement Techniques in a Cloud Data Center," *Procedia Computer Science*, vol. 78, pp. 491-498, 2016/01/01/ 2016, doi: <https://doi.org/10.1016/j.procs.2016.02.093>.
- [2]. E. Casalicchio, "Container Orchestration: A Survey," *Systems Modeling: Methodologies and Tools*, p. 14, 2018.
- [3]. J. Zhang, Z. He, H. Huang, X. Wang, C. Gu, and L. Zhang, "SLA aware cost efficient virtual machines placement in cloud computing," in *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*, 5-7 Dec. 2014 2014, pp. 1-8, doi: [10.1109/PCCC.2014.7017101](https://doi.org/10.1109/PCCC.2014.7017101).
- [4]. H. I.-K. E. A.-E. H. A. J. Forestier, "Network Issues in Virtual Machine Migration," *arXiv*, 2017.
- [5]. S. Shenai and M. Aramudhan, "A federated cloud computing model with self-organizing capability using trust negotiation," in *2017 International Conference on IoT and Application (ICIOT)*, 19-20 May 2017 2017, pp. 1-6, doi: [10.1109/ICIOTA.2017.8073605](https://doi.org/10.1109/ICIOTA.2017.8073605).
- [6]. Z. Tao *et al.*, "A Survey of Virtual Machine Management in Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1482-1499, 2019, doi: [10.1109/JPROC.2019.2927919](https://doi.org/10.1109/JPROC.2019.2927919).
- [7]. R. G. Alakbarov and O. R. Alakbarov, "Selection Virtual Machine in Mobile Cloud Computing," in *2018 9th International*

- Conference on Computing, Communication and Networking Technologies (ICCCNT), 10-12 July 2018 2018, pp. 1-4, doi: 10.1109/ICCCNT.2018.8493733.
- [8]. H. H. Jiangtao Zhang, Xuan Wang, "Resource provision algorithms in cloud computing," *J. Netw. Comput. Appl.*, vol. 64, p. 19, 2016.
- [9]. W. Lau, "Installing Third Party Software on Windows Azure – What are the options?," 2012. [Online]. Available: <https://www.red-gate.com/simple-talk/cloud/platform-as-a-service/installing-third-party-software-on-windows-azure-what-are-the-options/>.
- [10]. A. Tutorials, "Installing Software Packages." [Online]. Available: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-software.html>.
- [11]. K. A. SimiteOussama, "Impact Live Migration on Cloud Performance," *Smart Application and Data Analysis for Smart Cities (SADASC'18)*, 2018.
- [12]. B. S. Mishra P., Katal A., "Cloud Container Placement Policies: A Study and Comparison.," *Smys S., Senjyu T., Lafata P. (eds) Second International Conference on Computer Networks and Communication Technologies. ICCNCT* vol. 44, 2020, doi: [https://doi.org/10.1007/978-3-030-37051-0\\_58](https://doi.org/10.1007/978-3-030-37051-0_58).
- [13]. C. Academy, "Deploying Multiple Virtual Machines on Azure: Azure VM Scale Sets," *Cloud Academy* 2017. [Online]. Available: <https://cloudacademy.com/blog/deploying-multiple-virtual-machines-on-azure/>.
- [14]. I. Cloud, "Configuring virtual servers," *GitHub*, 2019. [Online]. Available: <https://cloud.ibm.com/docs/vsi?topic=virtual-servers-configuring-virtual-servers>.
- [15]. L. C. Prateek Sharma, Prashant Shenoy, Y. C. Tay, "Containers and Virtual Machines at Scale: A Comparative Study," presented at the Proceedings of the 17th International Middleware Conference Association for Computing Machinery, Trento, Italy, 2016.
- [16]. C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, "Cloud Container Technologies: A State-of-the-Art Review," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 677-692, 2019, doi: 10.1109/TCC.2017.2702586.
- [17]. S. Sultan, I. Ahmad, and T. Dimitriou, "Container Security: Issues, Challenges, and the Road Ahead," *IEEE Access*, vol. 7, pp. 52976-52996, 2019, doi: 10.1109/ACCESS.2019.2911732.
- [18]. N. G. Bachiega, P. S. L. Souza, S. M. Bruschi, and S. d. R. S. d. Souza, "Container-Based Performance Evaluation: A Survey and Challenges," in *2018 IEEE International Conference on Cloud Engineering (IC2E)*, 17-20 April 2018 2018, pp. 398-403, doi: 10.1109/IC2E.2018.00075.
- [19]. Docker, "The Modern Platform for High-Velocity Innovation," 2013. [Online]. Available: <https://www.docker.com/>.
- [20]. "Red Hat Ansible Automation Platform," 2012. [Online]. Available: <https://www.ansible.com>.
- [21]. Microsoft, "Create Linux or Windows virtual machines with your Azure free account," 2019. [Online]. Available: [https://azure.microsoft.com/en-us/free/virtual-machines/search/?&ef\\_id=EAIAIqObChMIgK6fpP-t5wIVw7HtCh3h9w9aEAAYASAAEgLuVvD\\_BwE:G:s&OCID=AID2000623\\_SEM\\_FcPIbhKp&MarinID=FcPIbhKp\\_377894685999\\_microsoft%20hyper%20v\\_e\\_c\\_\\_76767455506\\_kwd-300178430073&lnkd=Google\\_Azure\\_Brand&clid=CLfQ0Kj\\_recCFYtG1QodqSkBbw](https://azure.microsoft.com/en-us/free/virtual-machines/search/?&ef_id=EAIAIqObChMIgK6fpP-t5wIVw7HtCh3h9w9aEAAYASAAEgLuVvD_BwE:G:s&OCID=AID2000623_SEM_FcPIbhKp&MarinID=FcPIbhKp_377894685999_microsoft%20hyper%20v_e_c__76767455506_kwd-300178430073&lnkd=Google_Azure_Brand&clid=CLfQ0Kj_recCFYtG1QodqSkBbw).
- [22]. S. T. Help, "Top 10 Best Container Software In 2020," *Software Testing Help*, 2020. [Online]. Available: <https://www.softwaretestinghelp.com/container-software/>.
- [23]. V. R. U. D. Chitra Devi, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks," *The Scientific World Journal*, 2015.
- [24]. L. Wei, C. H. Foh, B. He, and J. Cai, "Towards Efficient Resource Allocation for Heterogeneous Workloads in IaaS Clouds," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 264-275, 2018, doi: 10.1109/TCC.2015.2481400.

- [25]. C. Colman-Meixner, C. Devellder, M. Tornatore, and B. Mukherjee, "A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2244-2281, 2016, doi: 10.1109/COMST.2016.2531104.
- [26]. F. Lombardi and R. Di Pietro, "Secure virtualization for cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1113-1122, 2011/07/01/ 2011, doi: <https://doi.org/10.1016/j.jnca.2010.06.008>.
- [27]. R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai, and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," *Journal of network and computer applications*, vol. 52, pp. 11-25, 2015.
- [28]. M. K. M. a. P. Murthy, Yasser and Sanjay, H. A. and Ameen, MohdNoorul, "Prediction Based Dynamic Configuration of Virtual Machines in Cloud Environment," 2012: IEEE Computer Society, pp. 124-128.
- [29]. D. Armstrong, Espling, D., Tordsson, "Contextualization: dynamic configuration of virtual machines," *Cloud Comp 4*, 17, 2015.
- [30]. N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, no. 3, pp. 175-185, 1992/08/01 1992, doi: 10.1080/00031305.1992.10475879.
- [31]. D. Coomans and D. L. Massart, "Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearestneighbour classification by using alternative voting rules," *AnalyticaChimicaActa*, vol. 136, pp. 15-27, 1982/01/01/ 1982, doi: [https://doi.org/10.1016/S0003-2670\(01\)95359-0](https://doi.org/10.1016/S0003-2670(01)95359-0).
- [32]. S. Sayad, "K Nearest Neighbors - Classification," 2009. [Online]. Available: [https://www.saedsayad.com/k\\_nearest\\_neighbours.htm](https://www.saedsayad.com/k_nearest_neighbours.htm).
- [33]. N. Anton Beloglazov, Manoel Campos da Silva Filho, TianZhang He, SarehFotuhi, William Voorsluys, "CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services," 2009. [Online]. Available: <https://github.com/Cloudslab/cloudsim>.
- [34]. a. singh, "Cloudlet in Cloudsim Simulation," 2019. [Online]. Available: <https://www.cloudsimtutorials.online/cloudlet-in-cloudsim-simulation/>.
- [35]. S. K. Sood and R. Sandhu, "Matrix based proactive resource provisioning in mobile cloud environment," *Simulation Modelling Practice and Theory*, vol. 50, pp. 83-95, 2015/01/01/ 2015, doi: <https://doi.org/10.1016/j.simpat.2014.06.004>.
- [36]. S. Kumaraswamy and M. K. Nair, "Intelligent VMs prediction in cloud computing environment," in *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 17-19 Aug. 2017 2017, pp. 288-294, doi: 10.1109/SmartTechCon.2017.8358384.
- [37]. F. Qiu, B. Zhang, and J. Guo, "A deep learning approach for VM workload prediction in the cloud," in *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 30 May-1 June 2016 2016, pp. 319-324, doi: 10.1109/SNPD.2016.7515919.
- [38]. M. R. Varun Barthwal, Rohan Verma, "Virtual Machines Placement Using Predicted Utilization of Physical Machine in Cloud Datacenter," presented at the International Conference on Advances in Engineering Science Management & Technology (ICAESMT) - 2019, Uttaranchal University, Dehradun, India, 2019.
- [39]. T. Veni and S. M. S. Bhanu, "Prediction Model for Virtual Machine Power Consumption in Cloud Environments," *Procedia Computer Science*, vol. 87, pp. 122-127, 2016/01/01/ 2016, doi: <https://doi.org/10.1016/j.procs.2016.05.137>.
- [40]. F. Farahnakian *et al.*, "Using Ant Colony System to Consolidate VMs for Green Cloud Computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187-198, 2015, doi: 10.1109/TSC.2014.2382555.
- [41]. D. K. Alzamil I., "Energy Prediction for Cloud Workload Patterns.," *Bañares J., Tserpes K., Altmann J. (eds) Economics of Grids, Clouds,*

- Systems, and Services. GECON 2016. Lecture Notes in Computer Science, vol 10382. Springer, Cham, 2017.*
- [42]. Z. Wu, X. Li, P. Garraghan, X. Jiang, K. Ye, and A. Y. Zomaya, "Virtual Machine Level Temperature Profiling and Prediction in Cloud Datacenters," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 27-30 June 2016 2016, pp. 735-736, doi: 10.1109/ICDCS.2016.62.
- [43]. M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "An Energy-Efficient VM Prediction and Migration Framework for Overcommitted Clouds," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 955-966, 2018, doi: 10.1109/TCC.2016.2564403.
- [44]. F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, and H. Tenhunen, "Utilization Prediction Aware VM Consolidation Approach for Green Cloud Computing," in *2015 IEEE 8th International Conference on Cloud Computing*, 27 June-2 July 2015 2015, pp. 381-388, doi: 10.1109/CLOUD.2015.58.
- [45]. T. Mahdhi and H. Mezni, "A prediction-Based VM consolidation approach in IaaS Cloud Data Centers," *Journal of Systems and Software*, vol. 146, pp. 263-285, 2018/12/01/ 2018, doi: <https://doi.org/10.1016/j.jss.2018.09.083>.
- [46]. M. Ferber, T. Rauber, M. H. C. Torres, and T. Holvoet, "Resource Allocation for Cloud-Assisted Mobile Applications," in *2012 IEEE Fifth International Conference on Cloud Computing*, 24-29 June 2012 2012, pp. 400-407, doi: 10.1109/CLOUD.2012.75.
- [47]. Y. Z. Zhijia Chen, Yanqiang Di, Shaochong Feng, "Self-Adaptive Prediction of Cloud Resource Demands Using Ensemble Model and Subtractive-Fuzzy Clustering Based Fuzzy Neural Network," *Computational Intelligence and Neuroscience*, p. 14, 2015.
- [48]. A. J. Younge, G. Von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," in *International Conference on Green Computing*, 2010: IEEE, pp. 357-364.
- [49]. M. Babu and S. Umamageswari, "OCRPs in dynamic resource allocation using virtual machines for cloud computing environment," in *Assistant Professor, PG Scholar: Citeseer*, 2014.
- [50]. E. Cuervo, Balasubramanian, A., Cho, D.-K., Wolman, A., Saroiu, S., Chandra, R. and Bahl, P., "Maui: Making Smartphones Last Longer with Code Offload," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys'10*, San Francisco, 2010, pp. 49-62. [Online]. Available: <https://scirp.org/reference/ReferencesPapers.aspx?ReferenceID=2055562>. [Online]. Available: <https://scirp.org/reference/ReferencesPapers.aspx?ReferenceID=2055562>
- [51]. Das, B. and KJ, M., 2017. Disability In Schizophrenia and Bipolar Affective Disorder. *International Journal of Psychosocial Rehabilitation*, 21(2).
- [52]. Elsass, P., Rønnestad, M.H., Jensen, C.G. and Orlinksky, D., 2017. Warmth and Challenge as Common Factors among Eastern and Western Counselors? Buddhist Lamas' Responses to Western Questionnaires. *International Journal of Psychosocial Rehabilitation*, 21(2).
- [53]. Weston, S.M., Martin, E.D., Shippen, M.E., Kraska, M.F. and Curtis, R.S., 2017. Parents with Serious Mental Illness Served by Peer Support Specialists. *International Journal of Psychosocial Rehabilitation*, 21(2).