A Neural Based Allocation Architecture Of Mobile Computing

Sudhir Kumar Sharma, Dr. Wiqas Ghai

Abstract: The computing architectures these days are getting sophisticated. The common person is aiming to get the services on the go. Researchers, therefore, envisage expanding cloud computing facilities to smart-phones limitations. The difficulty is that conventional smart-phone implementation designs do not promote the creation of apps that can integrate cloud-computing characteristics and require specific mobile cloud application designs. Scheduling, security, and load management is an essential part of the Cloud computing application architecture as well as mobile computing architecture. This paper presents an energy efficiency mobile computing architecture by using the Modified Best Fit Decreasing (MBFD) algorithm with Artificial neural Network (ANN) as a machine learning approach. The tasks are sorted using MBFD approach and the problem (Over-loading and under-loading) the server is resolved using ANN as a classification approach. The results shows that the tasks completed by mobile servers using ANN approach with less time and also required minimum energy to complete.

Index Terms: Mobile Cloud, Applications, Architectures, Scheduling, and Load Management, MBFD, ANN.

1 INTRODUCTION

Cloud computing is the important accessibility of computer system assets, particularly data accessing and calculating energy, without immediate effective customer leadership. The word is usually used to define data centers that are accessible over the Internet to many customers. Large clusters, dominant today, often have features dispersed from main computers over various places. If the link to the consumer is comparatively near, an apex server may be appointed. Cloud computation offers online computation, processing, facilities, and applications [1]. Besides, cloud computing promotes the reduction of investment costs, decouples facilities from the fundamental technology, and offers stability in aspects of energy provisioning.

1.1 Mobile Cloud Computing

Mobile cloud computation (MCC) can be described as a fresh distributed computing paradigm as an architecture, implementation, or method where data storage and handling moved from intelligent mobile technologies to distributed cloud computers. MCC can be explained as a new distributed computing paradigm as a design, application, or process where information processing and processing move from smart portable to distributed cloud facilities. The MCC provides many benefits for users at an organizational and community level and provided these benefits, and the MCC facilities are regarded to be the industry's growth engine [2]. As shown in Figure 1, an MCC can have different computing devices like mobile, laptop, etc.

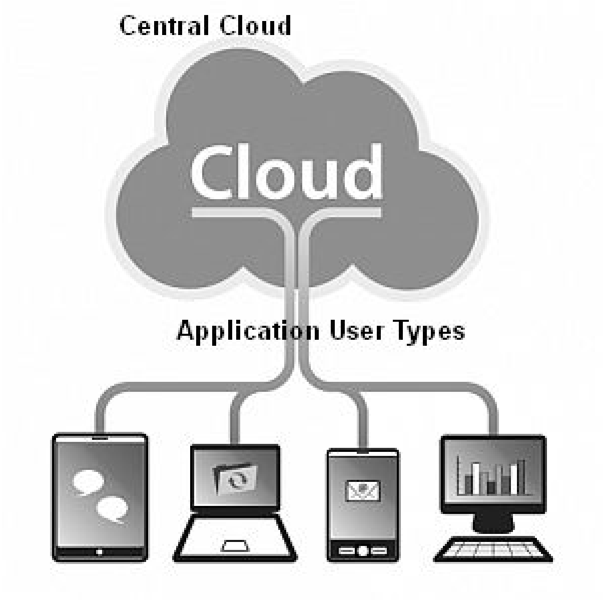


***Fig.1**. MCC Computation and Application*

Computation handling is a process that migrates energy-intensive calculation from a smart-phone to the memory-rich cloud or computer (called neighboring facilities). Cloud-based computing offloading improves application performance, decreases battery power utilization, and executes apps that are not to run owing to inadequate smart-phone funds. Besides, the cloud provides memory facilities that can be used to solve smart-phone memory limitations. Many apps currently occur for various fields with cloud assistance, such as trade healthcare education, social networks gaming file sharing and looking among others. We describe mobile cloud computing as integrating cloud computation software with portable apps to render mobile devices resource-full in aspects of computational power consumption, storage utilization, energy conservation, and context awareness. Mobile cloud computation is the result of interdisciplinary methods to mobile computing and cloud computing [3].

- *Sudhir Kumar Sharma is currently pursuing research scholar in Department of Computer application, RIMT University, MandiGobindgarh.*
- *Dr. Wiqas Ghai is currently working as an associate professor, Department of Computer Application, RIMT University, MandiGobindgarh*

1.2 Needs of Mobile Cloud Computing

There are following Needs of mobile cloud computing:

a) Flexibility: Mobile computing provides you access to use it anywhere you want and as much as you want.

b) Data Availability at all times: With this, you can access your data in different remote devices with any disturbance.

c) Cost efficiency: This system is very cost-friendly as there is no hefty charge linked to mobile cloud computing because it seems like the service is focused only on paying for what you are using.

d) Data back-up: The mobile cloud application gives you to back up your information on the cloud when it requires to be maintained safe or when the information is not in use, as you are continually generating new information on your device.

e) Data recovery: In the event of a catastrophe, you lose your critical information, the cloud implementation always enables you to retrieve your information from the cloud by pursuing an individual method. Regeneration of your information from any destination is possible if you are connected to the grid when you have enough storage capacity on your device [4].

1.3 Issues in Mobile Cloud Computing

a) Data Privacy: Most of the moment, the customer has sensitive cloud information, and there may be a break down in the network during the information stream that may add to data loss. It is essential to select the right service provider to ensure that your data safe at all times and in a different situation.

b) Connectivity: Whenever the network you want to use depends entirely on the internet connection, it is essential to see that the relationship is up at all times so that your cloud attachment is not affected which could affect the transition of your

1.4 Clone Based Mobile Computing

In Clone-Cloud, data protection and application piracy are of high interest from the view of clones. For example, if an opponent gets a clone of the smart-phone from either the cloud, the clone can be infinitely customizable along with the same smart-phone template. Therefore, the opponent may use the information of the copies and integrated apps that may contribute to information protection and software trafficking problems. Clone-Cloud's original parts are node, migratory, data storage station, profiler (vibrant profiler) and section analyzer. The migratory is accountable for both parties expelling, wrapping, commencing, and combining the thread countries.The benefit of this model is that the clone can be used as a buffer to recover information and apps when a smart-phone is damaged or demolished. Besides, Clone-Cloud increases the execution of cloud-based smart-phone applications by performing code analysis for software partitioning, bringing into account the expense and limitations of offloading. The server supervisor conducts the provisioning, picture synchronization, and manages the interaction between the moved nodes. Finally, the repository is accountable for recording request partitions. The findings indicate that CloneCloud-based apps have achieved efficiency improvements of 21.2 times in terms of execution moment. CloneCloud also promotes fine-grained thread-level migration, which is more useful as opposed to traditional suspend-migrate-resume processes. Considering the shortcomings, the system can only migrate at implementation stages where no original stack condition is gathered. Also, CloneCloud needs cost model created for each implementation under distinct partitions, where each partition is performed individually on the mobile device and cloud. Therefore, the implementation of portable phone partitions for price design creation may require additional power. Also, basic and fine-grained parallelization between the device and the clone is designed to fit all of the suggested enhancement types, which may be commodity-intensive in terms of bandwidth usage and energy consumption. Nevertheless, the writers suppose the cloud atmosphere is safe, which is not always the situation [5].

2 RELATED WORK

Akherfi et al. (2018) proposed the present offloading frameworks situation, computing offloading methods, and their primary critical problems are analyzed. It also discusses various significant parameters on the basis of which the considerations are applied, such as offloading technique and partitioning rate, and summarizes various problems in unloading mechanisms in the MCC domain that require further studies. The main issue in this paper is that they do not discuss the minimization of this issue [4]. Yang et al. (2018) propose a fresh FREDP (File Remote Key Encryption and Data Protection) method involving three-party communication between portable terminals, personal windows, and government servers. The portable terminal and personal clusters decrypt the cipher-text document to allow the portable terminal consumer to use the information. We also evaluate FREDP's safety using BAN. The FREDP meets the safety requirement and shows that FREDP achieves excellent efficiency after the simulation outcomes. However, they do not use the optimization algorithm due to which its efficiency is quite slow [6]. Zheng et al. (2018) propose a multi-agent probabilistic training algorithm to achieve the NE with a secure convergence rate (also analytically extracted). Conduct experiments to validate the efficiency of the suggested algorithm and assess its efficiency in a vibrant setting. The simulation outcome demonstrates that the match is equal to a measured crucial game that has at least one Nash Equilibrium (NE). The main disadvantage of this paper is that there is no improvement in the efficiency and accuracy shown in this paper [7]. Somula et al. (2019) proposed Healthcare apps that can be handled using a fresh cloudlet system to reduce handling time and provide sufficient safety for user information. The customer links to the accessible cloudlet and if the cloudlet does not provide the necessary funds or facilities. The customer will also redirect the cloudlet to private cloud to evaluate customer medical records, and the outcome after the distinct simulation of the suggested job indicates that the period of the method is minimized, but there is no any effect on the accuracy shown in this paper [8]. Arpaci et al.(2019) proposed a predictive model explaining the function of information management (i.e., retrieving, storing, sharing, and applying) methods of learners in anticipating their reactions to MCC use. In addition to a classical SEM-based strategy oriented on information gathered from 308 undergraduate learners, MCC facilities for instructional reasons and the research-validated the model by using supplementary machine learning algorithms. The technology used predicted developmental intentions with a consistency of more than 72 percent in most cases, but the time duration of the mechanism remains the same [9]. Sundararaj et al. (2019) proposed that the algorithm operates on a two-way MCC model with an offloading method that takes into account both the ' cloudlets ' and the audience ' cloud. 'The' cloud ' and the ' cloudlets ' are intended to predict clients operating hours in time limitation depending on the queue model. It also aims at optimally assigning tasks to manage the' cloudlets' load and reduce the average overall time for all tasks, and the result shows that the proposed automated system is highly accurate, but if they use the optimistic algorithm or classification the result will be improved [10]. Xu et al. (2019) proposedA block-chain Multimedia Workflow Cloudlet Management Method (MWSM). Block-chain is used to guarantee data integrity during the offloading process and NSGA-III (Non-dominated Sorting Genetic Algorithm III) is also used to implement the QoS improvement. ELECTRE (Elimination Et ChoixTradulsant la REaltite) is used to address the decision-making issues of the most appropriate scheduling strategies. After the simulation, the outcome indicates that the effectiveness and precision are greater, but the moment required [11]. Li et al. (2019) proposed a three-layer trust-enabled system structure model for mobile cloud computing technologies depending on a blurred extensive assessment technique. Because traditional methods can not adequately represent, the independent cooperation between mobile cloud organizations, a prototype scheme centered on the JADE multi-agent platform is introduced to assess the effectiveness of the suggested approaches. The simulation outcome shows that our approach improves the transaction success rate and user satisfaction, but accuracy is still a problem in this paper [12].

3 PROPOSED WORK

In this research, initially N number of nodes (mobile server) has been deployed in the cloud network. After that, tasks are assigned to the nodes based on the concept of MBFD algorithm. MBFD algorithm sorts the tasks as per their power level. The server that consumes less energy to complete the task is positioned at first and server that consumed highest energy to complete task is positioned at last. In this way, the tasks are handled by MBFD approach. But, the problem of under-loading and overloading the server is not overcome using MBFD approach, as using this approach; the tasks are allocated to the server without considering their loading or under-loading concept of tasks. To resolve this problem, we used ANN as a classifier, which distinguish between Over-loaded and Under-loaded server and hence assign tasks to that server, which is under-loaded. The detail description of the entire research work is provided in the following section.

3.1 SCHEDULING AND ALLOCATION POLICY

Scheduling is a big aspect when it comes to managing the incoming request at the cloud interface, whether it be a green cloud or the mobile cloud. Virtualization refers to creating instances from the existing physical hardware that last only until they are required. The instances are also called Virtual Machines (VMs). VMs share the load of the physical machine and release the resource as soon as the job is completed. When it comes to cloud, the VMs are managed by the cloud itself, and the VMs faces two concepts, namely VM allocation and VM migration. VM allocation process is being observed and modified by many researchers, including the creator of CLOUDSIM (Tool for the implementation of Cloud Architecture, http://www.cloudbus.org) Dr. Buyaa [2]. The algorithm used for sorting tasks in MCC is written below.

Algorithm 1: *Modified Best Fit Decreasing (MBFD)*

```
Input: host_( list),Virtual_Machine  list Output: Allocation
Virtual_(Machine_list ).arrange in decreasing utilization ( )
For each Virtual_Machine in Virtual_Machine list do
Minimum_(Required_Power)←Maximum_(Available_Power )
aloocated_host←null
for each Host in host list do
if Current_Host has enough_resource for Virtual_Machine
 then
power←estimate power (Host,Vm)
    If power<minimum_power then
Allocate host←host
Minimum_Required_i power←power
if Assigned host≠null then
Allocate VIRTUAL_MACHINE to Assigned Host
Return Allocation2
```

The MBFD algorithm has two constraints, namely the available resources of the Host and the demand of the Virtual Machine. The VMs are sorted on the base of the Host demand and then are processed according to the Availability of the Host. The highest demanding VM is allocated first followed by the lower demanding VMs. Every time a VM is allocated to a host, its resources are decreased by a certain amount, and that would be equal to the total demand of the Virtual Machine.

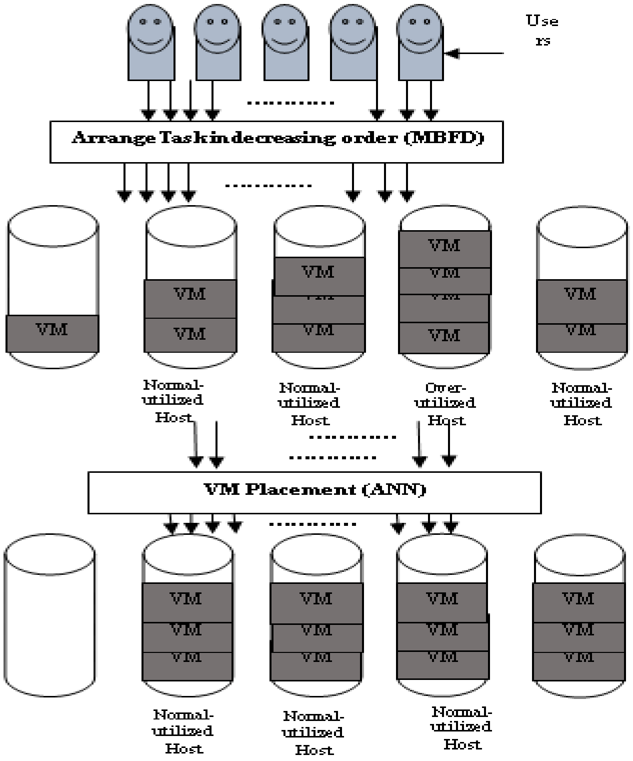
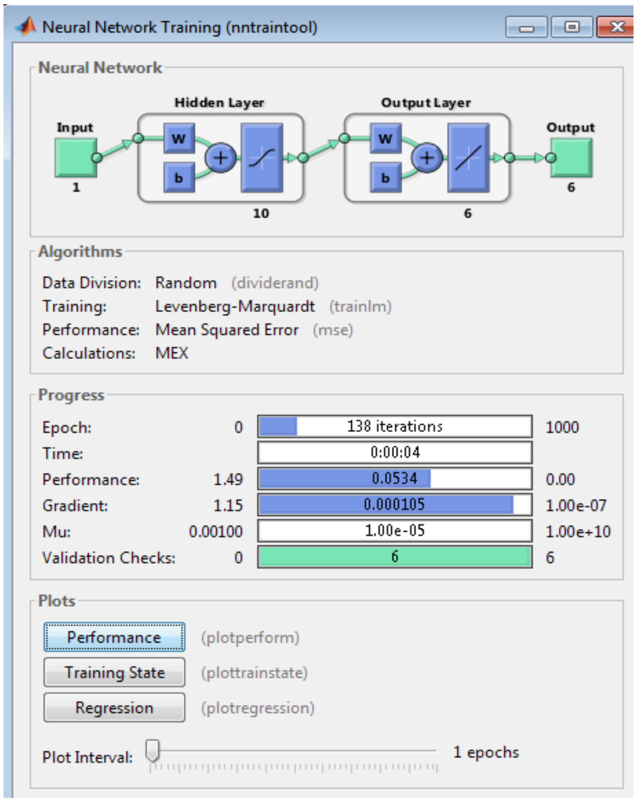


***Fig.2.** Flow of Proposed Work*

As shown in Figure 2, the hosts that is overloaded and under-loaded are identified by ANN as machine learning approach and then moved the tasks or the VMs towards those hosts so that the load can be managed as shown in Figure 2.
If the allocation process is not done well, the system faces migration issues, and hence, the VM is posted to another Physical Machine to maintain the flow of the scheduling. It is also possible that the VM gets migrated due to an overload of the Physical Machine. Based on the utility, VM migration is categorized as follows.

a) Migration due to overload: Demonstrates the poor management of load balancing
b) Migration due to a shortage of resources: Demonstrate the poor management of allocation.

The allocation policy also depends upon the type of demand, which is made by the user and the type of allocation which is being provided. Firstly, on the basis of ideal properties of VM, the proposed model is trained, which helps in final selection and placement of VM. Figure 3 illustrates the training screenshots, followed by ANN algorithm.



Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Epoch:	0	138 iterations	1000
Time:		0:00:04	
Performance:	1.49	0.0534	0.00
Gradient:	1.15	0.000105	1.00e-07
Mu:	0.00100	1.00e-05	1.00e+10
Validation Checks:	0	6	6

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

***Fig.3**. ANN Structure*

***Algorithm 2:** ANN*

```
1.  Start
2.  Call and set the ANN using VM properties as training data (T-Data), No. of VMs as a Target (TR) and Neurons (N)
3.  Set, Mobile Cloud-Net = Newff (T-Data, TR, N)
4.  Cloud-Net.TrainParam.Epoch = 1000
5.  Cloud-Net.Ratio.Training = 70%
6.  Cloud-Net.Ratio.Testing = 15%
7.  Cloud-Net.Ratio.Validation = 15%
8.  Cloud-Net = Train (Cloud-Net, T-Data, TR)
9.  Current VM = Properties of current VM in Mobile Cloud-Net
10. VM Characteristics = simulate (Mobile Cloud-Net, Current VM)
11. If VM Characteristics is valid and have power then
12. VM = Validated
13. Else
14. VM = Need correction because may be under or over loaded VM
15. End
16. Return: VM as a valid allocation
17. End
```

4 PERFORMANCE ANALYSIS

The recent architecture of system development in terms of cloud allocation and migration is monitored in lot of research articles mentioned in the related work section. The evaluation of these system architectures are done on the base of Quality of Service (QoS) parameters which may include the following.

1. SLA Violation: SLA stands for service level agreement. There is no such fixed definition of this element. It is a said or unsaid bond between the service provider and the service demander. It also depends upon the total number of assigned VMs to the total number of Physical Machines (PMs). As shown in Figure 2, increase in VM in the network results into low SLA-V when the number of PMs increases simultaneously. On the other hand, if the VMs are increasing and the PMs are stationary, the SLA-V increases.

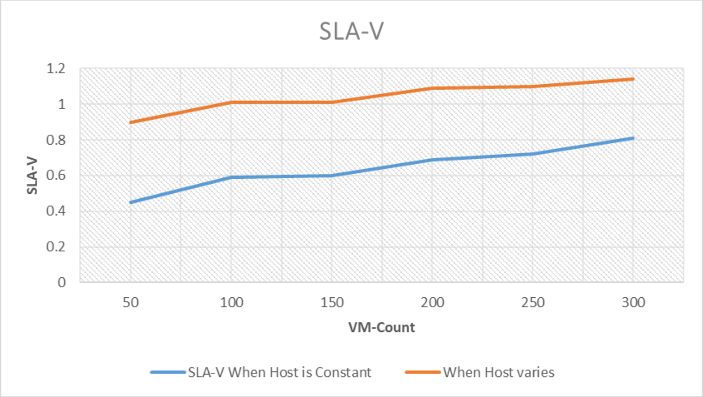


***Fig.4.** SLA V*

2. Job Completion Ratio (JCR): It is the ratio of the total number of Completed job over the total number of supplied jobs. Likely to be varying in the same scenario, this ratio also increases with the increase in the VM count if the PM count increases simultaneously. Figure 3 depicts the results for the same.

Other computation parameters that have been evaluated are energy consumption and completion time with respect to number of mobile users.

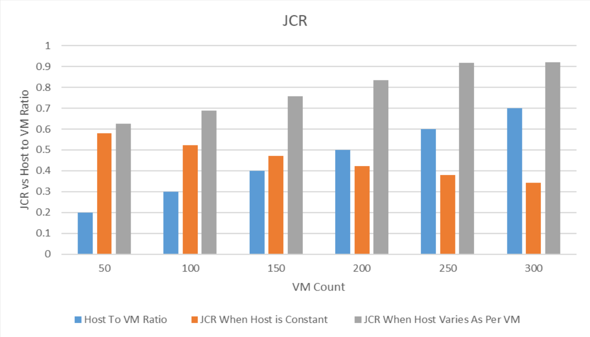


Fig.5. *JCR*

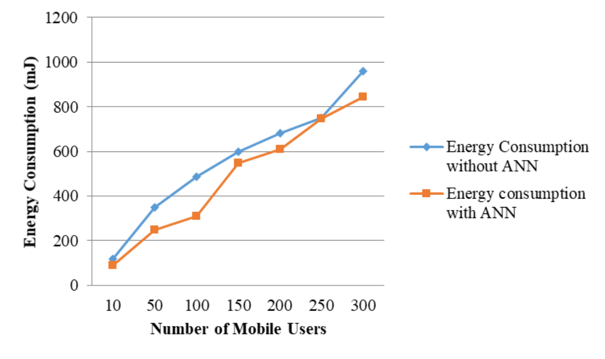


Fig.6. *Energy Consumption*

Figure 6 shows the graphical representation of energy consumed during the tasks allocation with varying number of mobile users. As seen from the figure that as the number of mobile users increases the energy consumption also increases. Using ANN approach the energy consumption has been reduced by great extent. The average of energy consumed by the mobile network without ANN and with ANN approach is 563.7 mJ and 485.7 mJ respectively. Therefore there is a reduction of 13.84 % while using ANN approach.

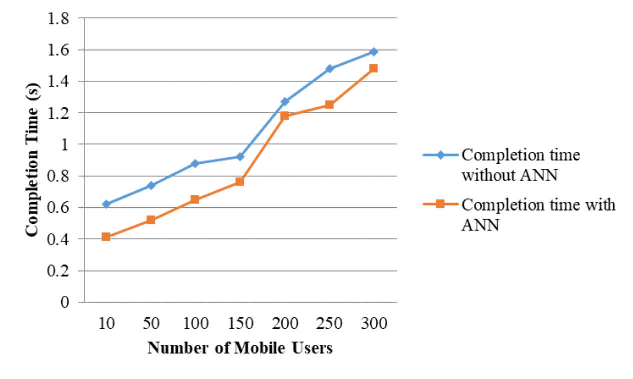


Fig .7. *Completion Time*

Figure 7 represents the complete time required by the mobile users to complete the allocated tasks. From the graph it is clear that when ANN approach is used to complete the task, the completion time is less. The average of completion time analyzed without ANN approach and with ANN approach are 1.07 s and 0.89 s respectively. Thus, using ANN approach task is completed with 16.82 % faster than traditional MBFD approach.

5 CONCLUSION AND FUTURE SCOPE

Mobile cloud technology seeks to empower portable users by offering a seamless and wealthy interface independent of the energy constraints of mobile devices. Although still in its infancy, in the future, portable internet technology could become the dominant model for portable apps. In this article, we gave a comprehensive study of present studies on portable cloud computing. We also provided various meanings of mobile cloud computing in the literature to highlight the motive for mobile cloud computing. We provided taxonomy of problems identified in this region and the methods that addressed these problems, concentrating on an operational level, end-user stage, level of delivery and implementation, safety, and context awareness. A reliable and fast mobile cloud computing based system has been presented using MBFD with ANN approach. The tasks have been assigned with more accuracy and also the problem of under-loading and over-loading has been resolved using ANN as a machine learning approach. There are countless fresh portable apps that a mobile cloud structure can allow when more funds can be rendered accessible to the mobile device (via the mobile cloud warehouse). The future could also investigate the ability of local portable clusters created from computer libraries in widespread systems in clothes, clothing, clocks, jewelry, furniture, and other everyday items, as such embedded computers will actually become more potent. Moreover, so the facilities, framework or implementation available as services will be of new forms: the technology could be a powerful significantly distributed set of cams on moving and mobile devices created ad hoc and measured to cover an event, or a collection of disseminated computers formed to calculate a job seamlessly from the user's smart-phone while the user is shopping.

6 REFERENCES

[1] Tucker, Rod, Kerry Hinton, and Rob Ayre. "Energy-efficiency in cloud computing and optical networking." European Conference and Exhibition on Optical Communication. Optical Society of America, 2012.

[2] Beloglazov, Anton, Jemal Abawajy, and Rajkumar Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." Future generation computer systems 28.5 (2012): 755-768.

[3] Fernando, N., Loke, S. W., &Rahayu, W. (2013). Mobile cloud computing: A survey. Future generation computer systems, 29(1), 84-106.

[4] Othman, Mazliza, Sajjad Ahmad Madani, and Samee Ullah Khan. "A survey of mobile cloud computing application models." IEEE Communications Surveys & Tutorials 16.1 (2013): 393-413.

[5] Fernando, Niroshinie, Seng W. Loke, and Wenny Rahayu. "Mobile cloud computing: A survey." Future generation computer systems 29.1 (2013): 84-106.

[6] Akherfi, Khadija, Micheal Gerndt, and Hamid Harroud. "Mobile cloud computing for computation offloading: Issues and challenges." Applied computing and informatics 14.1 (2018): 1-16.

[7] Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. Future Generation Computer Systems, 78, 964-975.

[8] Yang, Li, et al. "A remotely keyed file encryption scheme under mobile cloud computing." Journal of Network and Computer Applications 106 (2018): 90-99.
[9] Zheng, Jianchao, et al. "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach." IEEE Transactions on Mobile Computing 18.4 (2018): 771-786.
[10] Somula, R., Anilkumar, C., Venkatesh, B., Karrothu, A., Kumar, C. P., &Sasikala, R. (2019). Cloudlet services for healthcare applications in mobile cloud computing. In Proceedings of the 2nd International Conference on Data Engineering and Communication Technology (pp. 535-543). Springer, Singapore.
[11] Arpaci, Ibrahim. "A hybrid modeling approach for predicting the educational use of mobile cloud computing services in higher education." Computers in Human Behavior 90 (2019): 181-187.
[12] Sundararaj, V. (2019). Optimal task assignment in mobile cloud Sundararaj, Vinu. "Optimal task assignment in mobile cloud computing by queue based ant-bee algorithm." Wireless Personal Communications 104.1 (2019): 173-197.
[13] Xu, Xiaolong, et al. "Blockchain-based cloudlet management for multimedia workflow in mobile cloud computing." Multimedia Tools and Applications (2019): 1-26. Xu, Xiaolong, et al. "Blockchain-based cloudlet management for multimedia workflow in mobile cloud computing." Multimedia Tools and Applications (2019): 1-26.
[14] Li, Wenjuan, et al. "A Trust-Based Agent Learning Model for Service Composition in Mobile Cloud Computing Environments." IEEE Access 7 (2019): 34207-34226.