Cuckoo-Neural Approach for Secure Execution and Energy Management in Mobile Cloud Computing

Vishal[1]
Department of Computer Science
and Engineering, IKG Punjab
Technical University,
Kapurthala, Punjab, India

Bikrampal Kaur[2]
Department of Computer Science
and Engineering, Chandigarh
Engineering College, Landran, Punjab, India

Surender Jangra[3]
Department of Computer Science, GTBC,
Bhawanigarh, Punjab, India

***Abstract*—Along with an explosive growth in mobile applications and the emergence of the concept of cloud computing, in mobile cloud computing (MCC) has been familiarized as a potential technology for mobile users. Employing MCC to enable mobile users to realize the benefits of cloud computing in an environment friendly way is an effective strategy to meet today's industrial demands. With the ever-increasing demand for MCC technology, energy efficiency has become extremely relevant in mobile cloud computing infrastructure. The concept of mobile cloud computing offers low cost and high availability to the mobile cloud users on pay-per-use basis. However, the challenges such as resource management and energy consumption are still faced by mobile cloud providers. If the allocation of the resources is not managed in a secure manner, the false allocation will lead to more energy consumption. This article demonstrates the importance of energy-saving mechanisms in cloud data centers and elaborates the importance of the "energy efficiency" relationship to promote the adoption of these mechanisms in practical scenarios. The utilization of resources are being maximized by minimizing the energy consumption. To achieve this, an integrated approach using Cuckoo Search (CS) with Artificial Neural network (ANN) is presented here. Initially, the Virtual Machines (VMs) are sorted as per their CPU utilization using Modified Best Fit Decreasing Approach (MBFD). This suffers from the increase in Service Level Agreement (SLA) violation along with many VM migrations. If the migration is not done at an appropriate host, which can hold the VM for long, Service Level Agreement Violation (SLAV) will be high.**

***Keywords*—Mobile cloud computing; VM migration; energy consumption; SLA violation; VM selection; overloading; under loading**

I. Introduction

Mobile Cloud Computing (MCC) is the development of numerous internet-based technologies that allow mobile users to take advantage of cloud computing while using their mobile devices and achieve green computing [1,2]. The MCC technology mainly the combination of three techniques, including mobile computing (MC), mobile internet and cloud computing systems. By taking the advantages of all these techniques, MCC allows user to upload data and stored into the MCC cloud server [3]. The. MCC offered advantages like as increase the storage capacity and provide virus scan facility to enhance the safety as well as reliability of mobile device. It's also minimizes the loss of data risk for mobile users. This is possible by sending data to the mobile cloud [4]. Although, MCC facilitate mobile user with a number of advantages, but still facing some challenging [5]. In this modern era, the population of smart phone users has increased intensely by some billions (see Fig. 1). In addition to the advantages and features of MCC, some issues like resource limitation, energy consumption, and cost of computation on the mobile, network and server side occurs in MCC [6]. In previous years, mobile users used MCC only to transmit text messages and requested only text information from the cloud server. Gradually, the requirements have changed and now most users prefer to transmit video data over cloud instead of text and images [7]. Compared to text or image data, video data required more energy to upload as well as to access data from the cloud. The data transfer using video data over the network consumes more energy that the results of pool quality of services. Therefore, the minimization of energy becomes a serious problem to the researchers. This requires mechanism to expand the energy efficiency of MCC data centers while maintaining the specified level of operations [8].

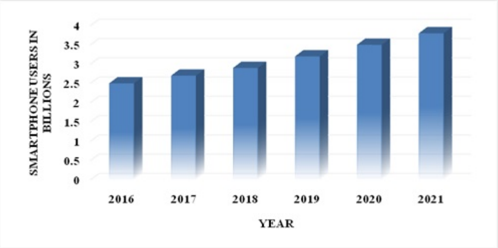

Fig. 1. World-Wide Growth of Smartphone Users

To minimize the power consumption, large scale computer system process is performed by using virtualization technology. Using this approach, a number of servers are consolidated into a single physical node, In other words one can say that the same copy of the software is created to perform several tasks such as storage, task allocation, to execute tasks and many more. This results in minimization of IT cost, whereas increasing the flexibility and efficiency of the system [9]. The recently emerging Mobile cloud computing paradigm employs virtualization and provides an on-demand resource on the internet to the mobile users.

This will allow mobile users to reduce the cost of maintaining their computing environment and move their computing needs onto the mobile cloud [10]. In the Virtual Machine (VM) migration policy, the VMs are migrated from one host to another in such a way that the idle host can be switched off and save the power consumed during its idle mode. The different parameters such as CPU utilization, overutilization and underutilization have been considered while migrating VMs [11]. Mainly two approaches are used to minimize energy named VM placement approach and VM consolidation approach. The VM placement is used while mapping VMs to PMs in a dynamic environment so that the utilization of resources can be optimized. After that, VM consolidation is employed to enhance the usage of resources by positioning under-loaded resources in idle state. This is performed by combining isolated VMs over a minimal active PMs by turning off the entire idle servers and hence saving the energy [12]. During the VM placement and consolidation process, the developer faces several challenges. The first one is related to time and resource-consuming nature of VMs. Another problem is related to the energy consumption of system. To resolve these problems, if one can turn off a few PMs, then the VMs are enabled to obtain resources at their peak time period. This results in the degradation of reliability and availability of the system that lower the desired QoS of mobile cloud service provider [13-14]. Therefore, to minimize the energy consumption by employing the VM consolidation technique, it leads to decrease the performance of the system. Hence, the SLA violation with the mobile user demand increases [15]. Therefore, to resolve the problem of VM placement and energy consumption, algorithms used such as Modified best Fit Decreasing (MBFD) and Artificial Neural Network (ANN). The aim of this paper is to reduce the power consumption and the SLA violation in the mobile cloud server.

Contribution of the paper is as follows:

1. To minimize energy consumption of resources.

2. To maximize the usage of resources in corporate with desired migration and allocation.

3. To provide satisfactory services to the cloud user.

The rest of the paper is organized as: the related work based on energy-aware and related to minimization of SLA violation is described in Section 2. The detailed description of the proposed work, along with their algorithms, is defined in Section 3. The results and discussion with comparison are demonstrated in Section 4. The overall conclusion of the entire article is presented in Section 5, followed by references.

II. Related Work

The VM allocation in the mobile cloud server is a two-way procedure. The first one is to admit new requests for VM placement on a stationary host. The second one is used to reallocate and find the best allocation for VM on a reliable host. The VM allocation has mainly two foremost goals. The first one is that the mobile cloud servers must utilize power in an efficient way and the other one is to provide maximum QoS to mobile cloud users. The present allocation of VM can be optimized through VM migration process that has been performed on the basis of pre-described constraints. The entire migration procedure is performed in three phases: a selection of best VM, destination host selection and selection of VM migration. The VM placement is an NP-hard problem that can be deliberated as a bin packing dilemma with different bin sizes in which bin represents the mobile cloud servers and VM is used as an object.
Wang et al. (2015) have formulated the problem occurs during the offloading process in MCC. As a result, energy consumption is minimized and increase the life time of mobile devices. The researchers have used polynomial-time optimal solution that works on bi-partile matching issues [16]. A traffic aware cross site as VM migration system has been proposed by Liu [17]. Using this model, multiple VMs are required to be migrated, which provides a wide range of cross site link that minimizes the count of successful VM migration. The authors have used heuristic algorithm to overcome this problem and obtained optimal results [17]. Halabi have provide security to the mobile cloud users by maintain the security service level agreement. By using the designed system, the preferences of mobile user as well as service providers has been evaluated during the process of resource allocation. By using Gale/Shapley algorithm, the problem is solved and rectified that delivers data with high stability and efficiency [18]. The algorithms such as first fit, next fit and random fit are the simplest VM allocation algorithms and they have not taken the capacity of resources into account while placing VMs. These techniques have not considered the capacity of the physical machine's resource consumption. Therefore, the power-aware algorithm has been initially presented. In which the problem of dynamic VM placement has been resolved with less power requirement and with smaller SLA violations. A p-mapper algorithm has been presented by Verma et al. [19] to resolve the power consumption and SLA violation problem, this technique mainly consists of three controlling units such as a power management unit, performance and migration management unit. This technique has performed both VM placement as well as VM reallocation to achieve an optimized allocation. The test results show that about 25 % of power has been saved using p-mapper algorithm compared to individual first fit and random fit placement algorithm. The energy in mobile cloud server can be saved by employing VM consolidation scheme.
Sharifian, S., & Barati (2019) have presented an integrated prediction scheme to address the issues scalability and resource allocation. Three level (3-L) wavelet transform has been used to decompose the time of workload series into multiple resolution of time and frequency scales. For classification of workload, support vector regression (SVR) has been used as prediction approach. The optimization of SVR parameters has been again refined using chaotic particle swarm optimization (PSO) algorithm as optimization approach. Again, the classification of high frequency data has been performed using generalized auto regressive conditional heteroskedasticity (GARCH) model. At last ensembled model has been applied to predict workload and the performance of the model has shown accuracy of 24.53 % of improvement. The drawback of this model is that the use of multiple approaches to identify workload is very complex process [20]. Mukherjee have presented a decision-making model whether the mobile data is offloaded or not. By using the presented strategy, if the power consumption of the device is decreases below the defined limit only then the data will be offloaded. The energy consumption of the device has been evaluated based on parameters such as computation time and deadline of the task. By using designed algorithm, the energy consumption has been reduced upto 3% to 32 % [21].

Gai et al.(2016) have presented a dynamic energy aware cloud let based MCC approach to optimize the power efficiently with reduced latency period[3]. This approach mainly focused to reduce the problem of energy consumption in MCC. Boukerche et al. (2019) have examined the presently used energy aware offloading schemes, architecture and balancing approaches that have been utilized to balance load and contributed towards green MCC[22]. Mohammed and Tapus (2017) have rectified the issues related to resource management and energy consumption in MCC. The designed model has worked into two phases initially; enthalpy has been calculated using task loss, transmission loss and delay in the second phase. The cuckoo search (CS) algorithm has been utilized to improve and arrange resources as per their priority that results in minimization of energy and cost [23].

Guo et al. [24] have presented an energy efficient based dynamic offloading scheme along with resource scheduling (edors) to minimize the energy consumption as well as decrease the task completion time. To overcome the problem, a new edors algorithm has been designed. The designed model comprises of n number of Smartphone's that are positioned in different regions each consist of an intensive mobile application that must be completed. In proposed model, the mobile application is divided into m number of tasks that have been completed by the mobile device in an MCC environment.

Tang et al. [25] have designed an MCC model specifically to resolve the issues of inadequate battery energy, less storage space. The system is mainly divided into two portions (i) the mobile user and (ii) the mobile cloud. To upload data into the mobile cloud (MC) an access point (AP) link is used. The upcoming tasks in MC have been arranged in a queue. Also, a control action has been performed to check the availability of the computing resources. Also, the tasks have been executed on the basis of the directed acyclic graph (DAG) model, in which the child task cannot be executed before the completion of the parent task. At last, the performance in terms of make span has been measured. The work has been completed for a few numbers of tasks that are only for 20 numbers of tasks, which is very less to check the efficiency of the designed network.

Problem Formulation

To manage energy in Mobile cloud data center is a challenging task. The around 70 to 80 percentage of power is misused by the cloud data center from total available power. Thus, proper monitoring and resource utilization need to be measured. To decide when and which VMs must be migrated to avoid the performance interferences. To select a correct destination host for migration is required for optimizing the VM migration because aggressive VM consolidation can cause performance interferences if the fluctuations on VM footprint rise at an unexpected rate which can lead to congestion. Thus, there is a need to select a correct host and reduce the congestion on the network to optimally utilize the network bandwidth.

III. Proposed Work

Usually, mobile users change their position due to the mobility of mobile nodes. However, the nodes in the system consume higher energy and hence restricted to their energy resources. While the user changes its location, the chances of energy consumption will be increased, that will result degradation of QoS parameters of the system. Therefore, the allocation of available resources to the mobile users with optimal power and improved QoS parameters becomes a great challenge to the MCC service providers. In this research, an optimized based ANN approach is used to overcome the challenge. The optimization has been performed by using CS as an optimization algorithm and distribution among loads has been determined using neural network algorithm.

This section deals with the methods and the techniques used to migrate the VMs with minimum energy and less SLA violation.

A. Power Model

In a mobile cloud environment, VMs, PMs, and Servers are the main sources of power consumption in terms of CPU utilization and MIPS of VMs and PMs are directly proportional to the CPU utilization which is calculated by the below formula:

$$P_{Consumed}(S) = Idle(P_{Consumed}) + S * (Max(P_{Consumed}) - Idle(P_{Consumed})) \tag{1}$$

Where:

S:It is servers active state CPU utilization

(P_{Consumed}(S)): It is the total consumption of power by the data centre.

Idle(P_{Consumed}): It is the consumption of power by servers in idle condition, which is considered as approximately 70% of Max(P_{Consumed}).

Max(P_{Consumed}): It is the maximum consumption of power by servers.

In the proposed model, CPU utilization of servers, VMs, and PMs may vary with respect to the time due to workload variation. So, the total consumption of power is a function of varying time based on demand and it is an integral of power consumed by active servers in the mobile cloud over a particular time period (Mukherjee et al. 2016). The formula of power calculation is written as:

$$P_{\text{Total}} = \int_0^t P_{\text{Consumed}}(S(t))dt \tag{2}$$

Mainly three different steps are performed by the proposed approach. Initially, the VMs are allocated to PMs using MBFD algorithm. Secondly, the VM selection using CS algorithm is performed from all over and under-utilized hosts. At the end, the selected VMs are placed on the host to minimize energy consumption and SLA violation as well (see Fig. 2).

B. VM Placement

The problem of VM allocation is subdivided into two segments: (i) To take requests from the users and then place the VMs on the host. (ii) To perform optimization of the present VM allocation. The first one issue is resolved by using the modified best fit decreasing order (MBFD) algorithm, in which the VMs are allocated to each host as per their CPU utilization. This is advantageous, when the resources are distributed in the heterogeneous mobile cloud environment and hence select the most efficient node to assigned (Gai et al. 2016). By using MBFD algorithm, all VMs are sorted in decreasing order as per their CPU utilization. These sorted VMs are then allocated to each host that offers the minimum power consumption after the allocation. This allows the heterogeneity of resources to be utilized by first selecting the most energy-efficient node. The algorithm in its pseudo code is written below. This algorithm faces n×m type of problem where n and m are the numbers of allocated VMs and the number of hosts, respectively.

MBFD Algorithm:

```
Start
Input: host list and VM list
Sort VM list as per CPU utilization in decreasing order
For each user
Calculate, Minpower<-max
Allocated VM<-[null]
For each VMs
Power, P<-Estimated Power (VMs, Users)
If P<mean(P) then
Allocate PM<- VM
Min(P)<-P
if Allocated server $\neq$ null then
Allocate VM
Sort VMs according to P
Return: Sorted VMs based on power utilization
End
```

C. VM Selection for Secure Placement and Better execution (Cuckoo Search)

In the optimization process, the present VM allocation is performed in two different steps as shown in Fig. 2: (i) firstly, those VMs are selected which are demanded to be migrated, (ii) Secondly, the selected VMs are allocated to the host by utilizing the ANN algorithm. We have applied artificial intelligence technique to know which VMs are integrated and what time. The Cuckoo search is a nature-inspired optimization algorithm which is applied here to resolve the n×m type problem of MBFD. This algorithm is inspired by a lazy bird named "Cuckoo". The bird cuckoo lays eggs on the nest of other birds that lay similar eggs. This feature increases the production of mature cuckoos, but sometimes the cuckoo's eggs are identified by the other host bird and hence removed from the nest. Usually, the increase in the number of alive eggs in the nest results into better benefits and that particular area is selected for optimization by the bird cuckoo. Therefore, the persisted cuckoo that survives and lay eggs on the other host's nest migrates or changes their position with maximum profit value. Here, the migration has been performed on the basis of the range covered by the bird to lay eggs. The process of laying eggs by the matured cuckoo is continued until the position with the maximum profit value can be attained.

Since CS is a fast and effective algorithm, it helps to resolve the optimization problem of VM selection with minimum energy. Initially, a few numbers of hosts are selected on a random basis. The features of hosts are presented in an array, which is known as habitat. The habitat of Cuckoo is represented by the following equation (3) which indicates the initial position of cuckoos.

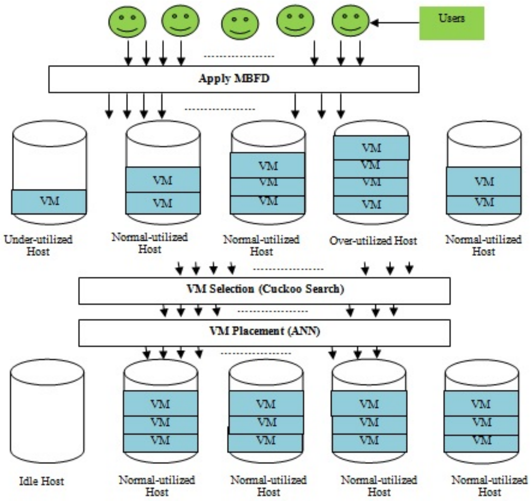

Fig. 2. Flow of Work

$$Habitat = [y_1, y_2, y_3, \ldots\ldots\ldots\ldots y_{\text{nvar}}] \tag{3}$$

After population initialization,(number of eggs represented by y_1,y_2,y_3,. . . .y_{nvar}), the next is to calculate the CPU utilization that is the profit value of hosts, it is computed using the following formula:

$$Profitvalue = f_{\text{fit}}(habitat) = f_{\text{fit}}(y_1, y_2, y_3, \ldots y_{\text{nvar}})) \tag{4}$$

$$f_{\text{fit}} = \begin{cases} TRUE & \text{if attained maximum profit value(}fs\text{ >fth)} \\ False & \text{otherwise} \end{cases} \tag{5}$$

Where f_S is the random change in egg and f^{th} is the threshold which should be achieved by host birds.

Using equation (4)(5), the maximum profit value is determined. Afterword's, each selected host is represented by a cuckoo and random eggs have been generated having upper and lower values. The eggs laying rage is calculated using formula written in equation (7) After population initialization, the nest is to calculate the CPU utilization that is the profit value of hosts, which is computed using the following formula:

$$Profitvalue = f_{\text{p}}(habitat) = f_{\text{p}}(y_1, y_2, y_3, \ldots y_{\text{nvar}})) \tag{6}$$

Using equation (6), the maximum profit value is determined. After that of each selected host represented by a cuckoo, random eggs have been generated having upper and lower values. The eggs laying rage is calculated using formula written in equation (7):

$$R_{\text{egg}} = \beta(I_{\text{eggs}})/(Tot_{\text{eggs}}(a_{\text{high}} * a_{\text{low}})) \tag{7}$$

Where, a_{high},a_{low} are the highest and lowest selected eggs for next generation. The I_{eggs} and Tot_{eggs} are the number of current cuckoo's eggs and total number of eggs, respectively. Now, based on the above-defined egg-laying range, the cuckoos lay eggs randomly, this concept is shown in Fig. 3. In Fig. 3, the red star denotes the random egg lying range, which is an initial cuckoo nest and the other random nests are represented by different color stars. It is assumed that a single egg consists of each individual host and then profit is determined. If the count of cuckoos is higher than the maximum available count of cuckoos, then the cuckoos with low profit are evacuated so that one can reach to the maximum cuckoo number. For this, the CS algorithm has used, K means clustering policy. Therefore, the set of cuckoos with maximum profit value has been selected and considered as a final point around which cuckoo has to revolve around. The cuckoos have not gone directly to the destination, but have moved with a deviation of (θ) the radius with an angle between -60° to $+60^\circ$. After little iteration, the larger portion about 95% is moved towards the best habitat that provides the maximum profit value. The following of the above-mentioned process helps CS algorithm to appropriate selection of the host to which VMs can be migrated. Now, the next process is to migrate VM from the over-utilized host to the underutilized host. It utilizes the space properly and hence leads to energy consumption.

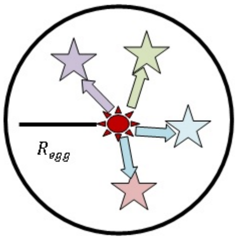

Fig. 3. Egg Laying Range [22]

Cuckoo Search Algorithm:

```
1. Start
2. Initialize CSA operators and parameter – Iterations (ITR) – Number of Egg (EN) – Number of Variables (Nvar)
3. Calculate Size of VM, Si<-Size (VM)
4. Fitness function $f_{\text{fit}}$
```

$$f_{\text{fit}} = \begin{cases} TRUE & \text{if attained maximum profit value}(fs > \text{fth}) \\ False & \text{otherwise} \end{cases}$$

Where fs is the random change in egg and f^{th} is the threshold which should be achieved by host birds

```
5.For each ITR & $S_i$ in respect of j
6.fs=$\sum_{j=1}^{s}$VM $_{\text{energy}}$=$E_N$
7.$f_{\text{th}}$=$\sum_{j=1}^{s}$Energy(VMs)/$s_j$
8. $f_{\text{fit}}$ = fitness function // which is define above
9. $VM_{\text{allocation}}$= CSA(EN,Nvar,$f_{\text{fit}}$ )
Return: Allocated VMs
10 End
11 Return: Allocated VMs
12 End
```

D. VM Validation to Cross Check the Allocation

Based on the received optimized properties of host, ANN is trained to find out over-utilized and under-utilized hosts. The trained structure of ANN is shown in Fig. 4 consisting of three-layered architecture that predicts the load based on the error estimation. The problem of over and underutilized hosts is resolved using the ANN technique. Here, a number of jobs are provided at the input layer on the basis of jobs properties such as CPU utilization RAM, ANN is trained. In the hidden layer, a number of neurons are passed to hidden layer in order to adjust the properties according to the weight. Hence, the output in terms of load prediction for over and underutilized host is provided. Since the over and under-utilized hosts have not provided services to the entire incoming requests from the user. This increases response time and hence leads to SLA violation. Therefore, for every mobile cloud service provider, it becomes necessary to manage over and under hosts. To overcome this situation, the ANN algorithm is used. For the CPU utilization of each and every host that has been determined using equation-5, ANN has been trained and then over or underutilized host is determined.

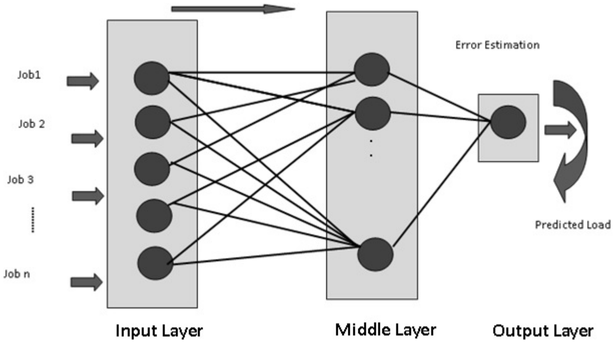

Fig. 4. ANN Structure

ANN Algorithm:

```
1 Start
2 Initialize ANN and define the essential parameters using VM properties as training data (T-Data), No. of VMs as a Target ($T_R$) and Neurons (N).
3 Set, Mobile cloud-Net = Newff (T-Data, $T_R$, N)
```

```
4 Mobile cloud-Net.TrainParam.Epoch = 1000
5 Mobile cloud-Net.Ratio.Training = 70%
6 Mobile cloud-Net.Ratio.Testing = 15%
7 Mobile cloud-Net.Ratio.Validation = 15%
8 Mobile cloud-Net = Train (Mobile cloud-Net, T-Data, $T_{\mathrm{R}}$)
9 Current VM = Properties of current VM in Mobile cloud-Net
10 VM Allocation=simulate(Mobile cloud-Net,Current VM)
11 If (VM Allocation = True)
12 VM = Not migrated
13 Else
14 VM = Need of migration
15 End
16 Return: VM as a allocation and migrated list of VMs
17 End
```

E. New Vendor Registration

When the cloud is expanding, new vendors will definitely aim to join the existing cloud management architecture to share the profit. In such a scenario, poor vendor selection will always lead to high energy consumption and more SLA violation. The proposed work incorporates Lagrange's interpolation for the appropriate placement of a new PM into the existing cloud architecture. The architecture of interpolation is as follows. The paper proposes two hypotheses for the registration of new node.

Null Hypothesis: The node is capable of serving the Cloud network.

Alternate Hypothesis: There are issues with the registration of node in the network and will consume high energy in the network with a large number of violations.

Proposed work aims to prove the null hypothesis worng and uses The Lagrange polynomial S(X) containing $degree \leq (n-1)$ that demands n number of SN containing public shares of network with co-ordinates (x_1,y_1=f(x_1)),(x_2,y_2=f(x_2)),...... (x_n,y_n=f(x_n)) is given by

$$S(X) = \sum_{k=0}^{n} P_{\mathrm{k}}(X) \tag{8}$$

Where P_{k} is given by

$$S(x) = Y_{\mathrm{k}} \frac{(x - x_{\mathrm{l}})}{(x_{\mathrm{j}} - x_{\mathrm{l}})} \tag{9}$$

where $l \geq 1$ and $l \leq n$ and $l \neq k$

If written explicitly for n=3 nodes,Lagrange polynomial is represnted as follows:

$$S(X) = Y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + Y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + Y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} \tag{10}$$

The separate polynomial can also be formulated as per Szego(1975) which was later called Lagrange's fundamental interpolation:

for first node

$$S(X_1) = Y_1 \frac{x_2 * x_3}{(x-x_2)(x-x_3)} \tag{11}$$

for Second node

$$S(X_2) = Y_2 \frac{x_1 * x_3}{(x-x_1)(x-x_3)} \tag{12}$$

for third node

$$S(X_3) = Y_3 \frac{x_1 * x_2}{(x-x_2)(x-x_3)} \tag{13}$$

The key which is generated by the integration of separate polynomials is represented as

$$G_{\mathrm{K}} = \sum_{k=0}^{n} S(K) \tag{14}$$

If G_{K} matches the network key only then the null hypothesis is proved to be wrong.

Pseudo Code for Share Verification:

```
1) Order=2; // Interpolation order
2) MY$_{\mathrm{VALUE}}$=[ ] // Initializing the key values to be Empty
3) For i =1:3 // For 3 nodes
a)counter=1;
b)Current$_1$=Node$_{\mathrm{IDI}}$ // Taking the first node as initial reference
4) for j=1:Nodes;
a)Current=Vehicles$_{\mathrm{j}}$; // For each interpolation, there would be 2 Rest Nodes
b) if Current1 $\approx$ Current // If nodes are not same
c) Rest(counter)=current;
d) counter=counter+1;
e) End If 5) End For 6) Deno=0 7) Deno=Current$_1$-Rest$_1$*Current$_1$-Rest$_2$// The denominator value
8) Num=Rest$_1$*Rest$_2$
9) My$_{\mathrm{value}}$ [i]=Num/Deno
10) Shared$_{\mathrm{key}}$=Share$_{\mathrm{Current1}}$*My$_{\mathrm{value}}$[i]
11) End For
```

In order to keep the data secure on the network, this paper also presents a dual encryption algorithm which stores the information of the migration and other attributes of the physical machine and virtual machine. This makes the communication secure and more reliable. The data security is achieved with the involvement of an advanced encryption. The AES algorithm employs three important parameters, namely, key length, number of rounds and the block size. During the encryption processes, bytes in the form of cipher state are used where input bytes are processed to state array that result into output bytes. Additionally, all round keys are also generated. The encryption steps involved in AES consist of five main operations described in following algorithm.

Advanced Encryption Standard (AES) Algorithm:

```
1. Input: $Pt_b$ // $Pt_b$ plain text box
$S_k$ is secret keys and $N_r$ is number of rounds
2. Determine State
$state_{var}$=($Pt_b$, $S_k$) //state variable
3. AddRoundKey($state_{var}$,$S_k$(0))
4. $For_{each}$ I in ($N_r$-1)
    a. SubBytes($state_{var}$) // performs byte substitution transformation
    b. ShiftRows($state_{var}$) //performs shift transformation to rows except first row
    c. MixColumns($state_{var}$) //performs MixColumns transformation
    d. AddKey($state_{var}$,Key  textsubscript i//Adds a RoundKey to the $state_{var}$
$End_{for}$
5. SubBytes($state_{var}$)
6. ShiftRows($state_{var}$ )
7. AddRoundKey($state_{var}$,$Key_{nr-1}$ )
8. Return: ($state_{var}$) // AES state variable
```

The above encryption algorithm performs the secure VM allocation. However, decryption is performed by reversing the above operations. A small set of example is demonstrated in Table I. Once the encryption is performed, a random shuffling order is created in order to rotate the generated index structure. The pattern of rotation is stored in a semi- storage cache memory which is used to decrypt the pattern once required. The cache memory is refreshed once a new combination is generated.

TABLE I. VM MIGRATION AND ENERGY CONSUMPTION WITH IMPLEMENTATION OF AES ENCRYPTION

VM ID	PM Number	Migrated To	Energy Consumption (KWh)
22	1	4	0.78
15	2	7	0.32
8	5	9	0.45
32	12	15	0.57
23	6	10	0.38
18	3	11	0.76

IV. Results and Discussion

During the test, the parameters such as energy consumption, SLA violation and number of VM migrations are evaluated and compared with the existing techniques. Table II represents the computed results simulated in CloudSim tool. The table shows that energy consumption is minimal as compared to the individual MBFD approach. Also, the numbers of migrations are less, which indicates that the application runs at a higher speed. Due to this, the mobile cloud data center consumes less energy to execute the requested application from the user. Also, the decrease in the number of hosts and SLA violations results in minimized energy consumption.

Fig. 5 shows the numbers of hosts to which number of VMs are allocated with MBFD integrated with CS as well as ANN techniques named as MFCSNN. The results show that with increase in the number of VMs, the number of hosts also increases but the allocation rate is less while utilizing the optimization algorithm with artificial machine learning technique.

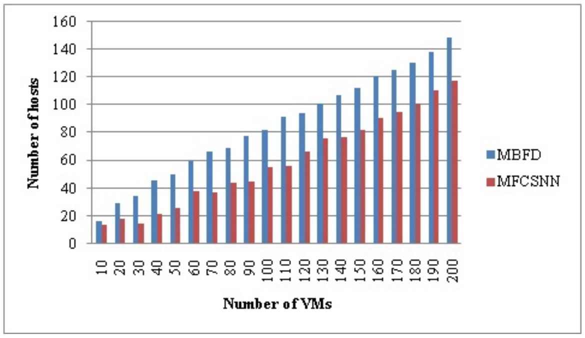

Fig. 5. Number of hosts vs VMs

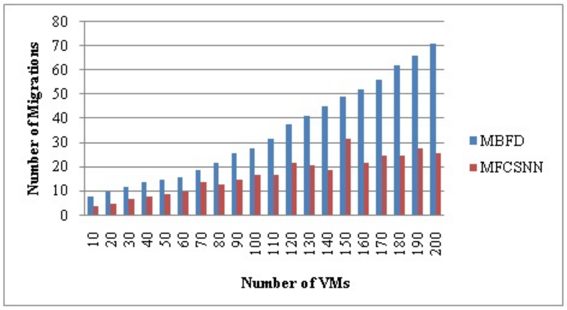

Fig. 6. Number of Migrations vs VMs

Fig. 6 indicates the graph for a number of migrations performed while executing the application using 200 numbers of VMs. The graph indicates that the number of migrations using the proposed MFCSNN approach is less compared to the existing MBFD approach. This is due to the proper selection of host. It also provides applications to the host, which is

TABLE II. COMPUTED RESULTS

Number of VMs	Number of hosts		Number of Migrations		Energy Consumption (KWh)		SLA Violation		Average Response Time (s)	
	MBFD	MFCSNN	MBFD	MFCSNN	MBFD	MFCSNN	MBFD	MFCSNN	MBFD	MFCSNN
10	16	14	8	4	0.3	0.2	9	7	1021	1011
20	29	18	10	5	0.4	0.3	8	6	1123	1094
30	35	15	12	7	0.5	0.25	11	10	3751	2574
40	46	22	14	8	0.7	0.48	13	11	5697	4875
50	50	26	15	9	0.7	0.38	14	9	7258	6527
60	60	38	16	10	0.9	0.65	15	10	7236	6851
70	67	37	19	14	1.0	0.87	18	13	8536	7142
80	69	44	22	13	1.3	0.85	22	15	8827	7892
90	78	45	26	15	1.4	1.02	24	19	9364	8538
100	82	55	28	17	1.4	1.09	25	21	10415	9456
110	92	56	32	17	1.7	1.14	27	24	11425	10752
120	94	67	38	22	1.9	1.15	29	26	12578	11364
130	101	76	41	21	2.2	1.35	31	27	12694	11687
140	107	77	45	19	2.3	1.48	35	30	12831	12015
150	112	82	49	32	2.4	1.78	39	34	13984	12675
160	121	91	52	22	2.9	1.75	44	42	14057	13624
170	125	95	56	25	3.1	2.05	46	43	14267	13965
180	131	101	62	25	3.4	2.38	48	45	14428	14025
190	138	111	66	28	3.6	2.67	50	47	14495	14270
200	149	118	71	26	3.8	2.66	53	50	14567	14396

under loaded and hence saves the energy consumption, which is unnecessarily utilized by the ideal host. The average VM migrations performed using MBFD and proposed MFCSNN approach are 32.15 and 16.95 respectively. The average value is considered as the mean of number of migrations obtained for different number of VMs varies from 10, 20, 30, 40 . . . .200 Hence, approximately 47.28 % reduction is attained by utilizing the CS with the ANN approach.

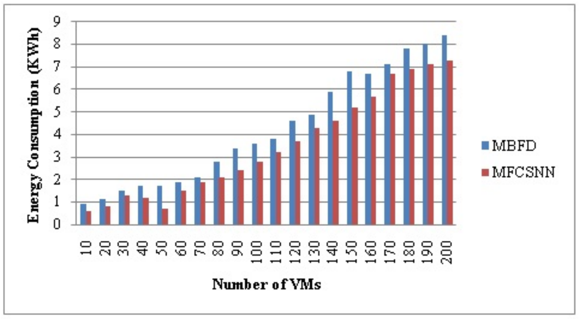

Fig. 7. Energy Consumption vs VMs

Fig. 7 shows that the energy consumption is the main parameter which must be minimized to obtain a green computing environment. It has been known that the ideal computer consumes about 70 % of total power without performing any operation. Therefore, it must be necessary to turn off those hosts which are not performing any operation. To achieve this goal, we have used ANN algorithm to distinguish among the ideal, underutilized and over-utilized host. Therefore, applications are forwarded to that host which is under-utilized and hence save energy. Using the proposed MFCSNN approach and simple MBFD approach, the average energy consumed is computed as 4.23 KWh and 3.5 KWh respectively. Thus, there is a reduction of about 17.26 %.

The SLA violation is an essential parameter that must be fulfilled by mobile cloud providers to ensure better delivery of mobile cloud services to their users. Along with the energy saving policy, it is essential to meet the SLA parameter of cloud. It indicates the agreement between the user and the cloud service provider. The higher SLA indicates better services are provided to the cloud user by the cloud providers and vice versa. The graph drawn in Fig. 8 indicates that the proposed technique provides a better series to their mobile cloud users. The average of SLA violation obtained for MBFD and proposed MFCSNN approach are 28.05 and 24.45, respectively. Therefore, there is a deterioration of 12.83 % while utilizing the CS with the ANN approach. The results based on the average response time with respect to the number of VMs are shown in Fig. 9. From this figure, it is clear that the designed MFCSNN algorithm outperforms as it required less time for migrating VMs to the host. Using traditional approach, the state of the jobs is restored if appropriate resources are not allocated and later allocate jobs whenever find available resources. But by using the proposed approach, time has been saved and therefore according to the demand, the resources are allocated. In addition to this, the computation overhead is also calculated for a new vendor to be added to the network. It is observed that when a new vendor is added into the network, every node not always provided the best outcome, which is desired. Hence to overcome the failure, a new vendor is also searched. Table IV presents the

TABLE III. COMPARATIVE VALUES

.	Energy Consumption(KWh)	.	Number of Migrations
MFCSNN	Karda and Kalra (2019) [23]	MFCSNN	Karda and Kalra (2019) [23]
3.5	5.59	16.95	40

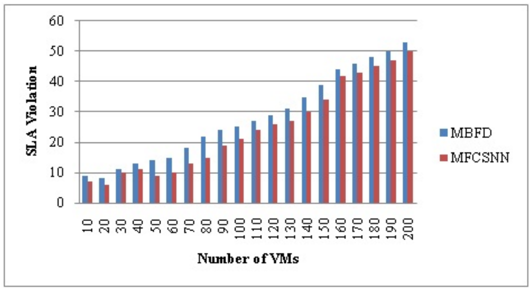

Fig. 8. SLA Violation vs VMs

TABLE IV. COMPUTATION OVERHEAD

Total registration of vendor	Computation Overhead
10	0.4
20	0.33
30	0.28
40	0.24

computation overhead, which is calculated by using equation 15.

The unsuccessful vendor is calculated by the observation of power consumption. If the node is consuming power which is higher than that of the average power consumption of the network for a given time interval, the vendor is unsuccessful.

To Show the comparative analysis comparison with the proposed and existing technique is presented in Table III. Here, the comparison of proposed MFCSNN approach is compared with the BAT and NN approach presented by Akki and Vijayarajan [8].

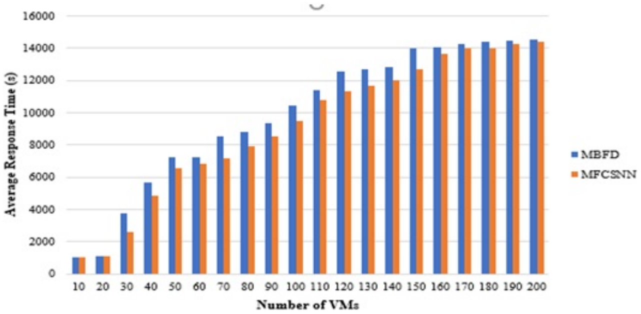

Fig. 9. Average Response Time

$$CO = \frac{(Total\ unsuccessful\ vendors)}{(Total\ added\ vendors)} \tag{15}$$

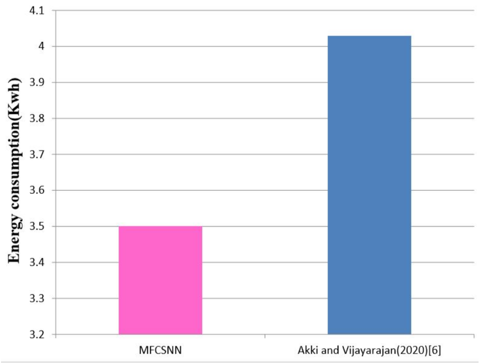

Fig. 10. Comparison of Energy Consumption

Fig. 10 and Fig. 11 represent the comparison graph drawn from energy consumption and number of migrations as given in Table III. The comparison has been made with the existing work performed by Akki and Vijayarajan (2020) [8] using BAT with ANN approach to optimize the power by selecting the best value among the available values of the generated solution. The aim of the researchers is to route mobile users to the server without interruption in services as well as with minimum power consumption. From the graph, it is clear that the integrated technique of CS with ANN performs well compared to an individual optimization scheme. Thus, there is an improvement of about 13.15 % and 8.23 % in the energy consumption, and the number of migrations respectively. This is because of the advantages of CS algorithm, as CS works on three parametric values. Therefore, the probability to providing optimized data is high. Also, the CS is able to detect a very small change in the result value so that appropriate solution can be obtained with small optimized time.

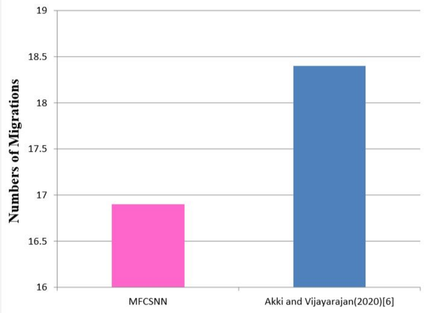

Fig. 11. Comparison of Number of Migrations

V. CONCLUSION

Power consumption has become a critical problem in the mobile cloud computing environment. The continuous growth in power consumption produces carbon dioxide. It has a negative impact on the green environment and hence raises the operating costs. In near future, there will be many scopes to improve load balancing by various techniques to minimize energy consumption within a secure mobile cloud computing environment.

In this article, we have suggested a load balancing technique called MFCSNN to minimize energy consumption in a mobile cloud computing environment. In addition to this, Lagrange's interpolation is incorporated with the existing modification of allocation and migration scheme for a new physical machine registration in the network. The parameters such as energy consumption, VM migration, number of hosts, and SLA violations are evaluated. As per the test results performed with MBFD and proposed MFCSNN, the performance of the designed system has been improved to offer secure VM allocation using AES. Hence it has been analyzed that the proposed model is a green solution with the minimum generation of carbon dioxide. Also, in order to show the effectiveness of the proposed method, comparison with the existing technology has also been presented. Here, as compared to the existing BAT -ANN approach, the proposed approach has minimized energy consumption up to 13.15 %.

REFERENCES

[1] Sabharwal, M., Agrawal, A., &Metri, G. (2012). Enabling green it through energy-aware software. IT Professional, 15(1), 19-27.

[2] Bonino, D., De Russis, L., Corno, F., & Ferrero, G. (2013). JEERP: energy-aware enterprise resource planning. IT Professional, 16(4), 50-56.

[3] Gai, K., Qiu, M., Zhao, H., Tao, L., &Zong, Z. (2016). Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. Journal of Network and Computer Applications, 59, 46-54.

[4] Ge, Y., Zhang, Y., Qiu, Q., & Lu, Y. H. (2012). A game theoretic resource allocation for overall energy minimization in mobile cloud computing system. In Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design (pp. 279-284).

[5] Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. Wireless communications and mobile computing, 13(18), 1587-1611.

[6] Eshratifar, A. E., &Pedram, M. (2018, May). Energy and performance efficient computation offloading for deep neural networks in a mobile cloud computing environment. In Proceedings of the 2018 on Great Lakes Symposium on VLSI (pp. 111-116).

[7] Jo, B., Piran, M. J., Lee, D., & Suh, D. Y. (2019). Efficient computation offloading in mobile cloud computing for video streaming over 5G. Comput. Mater. Contin, 61, 439-463.

[8] Akki, P., &Vijayarajan, V. (2020). Energy Efficient Resource Scheduling Using Optimization Based Neural Network in Mobile Cloud Computing. Wireless Personal Communications114, 1785–1804

[9] Akki, P., &Vijayarajan, V. (2019). An efficient system model to minimize signal interference and delay in mobile cloud environment. Evolutionary Intelligence, 1-9.

[10] Psannis, K. E., Stergiou, C., & Gupta, B. B. (2018). Advanced media-based smart big data on intelligent cloud systems. IEEE Transactions on Sustainable Computing, 4(1), 77-87.

[11] Shiraz, M., Abolfazli, S., Sanaei, Z., &Gani, A. (2013). A study on virtual machine deployment for application outsourcing in mobile cloud computing. The Journal of Supercomputing, 63(3), 946-964.

[12] Shiraz, M., &Gani, A. (2012). Mobile cloud computing: Critical analysis of application deployment in virtual machines. In Proc. int'l conf. information and computer networks (ICICN'12) (Vol. 27).

[13] Abd, S. K., Al-Haddad, S. A. R., Hashim, F., Abdullah, A. B., &Yussof, S. (2017). Energy-aware fault tolerant task offloading of mobile cloud computing. In 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud) (pp. 161-164). IEEE.

[14] Krishna, P. V., Misra, S., Nagaraju, D., Saritha, V., &Obaidat, M. S. (2016). Learning automata based decision making algorithm for task offloading in mobile cloud. In 2016 International Conference on Computer, Information and Telecommunication Systems (CITS) (pp. 1-6). IEEE.

[15] Yadav, R., & Zhang, W. (2017). MeReg: Managing energy-SLA tradeoff for green mobile cloud computing. Wireless Communications and Mobile Computing.

[16] Wang, X., Wang, J., Wang, X., & Chen, X. (2015). Energy and delay tradeoff for application offloading in mobile cloud computing. IEEE Systems Journal, 11(2), 858-867.

[17] Liu, J., Li, Y., Jin, D., Su, L., & Zeng, L. (2015). Traffic aware cross-site virtual machine migration in future mobile cloud computing. Mobile Networks and Applications, 20(1), 62-71.

[18] Halabi, T., Bellaiche, M., &Abusitta, A. (2019). Toward secure resource allocation in mobile cloud computing: A matching game. In 2019 International Conference on Computing, Networking and Communications (ICNC) (pp. 370-374). IEEE

[19] Verma, A., Ahuja, P., &Neogi, A. (2008) pMapper: power and migration cost aware application placement in virtualized systems. In Proceedings of the 9th ACM/IFIP/USENIX International conference on middleware. pp. 243-264. Springer-Verlag New York, Inc

[20] Sharifian, S., &Barati, M. (2019). An ensemble multiscale wavelet-GARCH hybrid SVR algorithm for mobile cloud computing workload prediction. International Journal of Machine Learning and Cybernetics, 10(11), 3285-3300.

[21] Mukherjee, A., & De, D. (2016). Low power offloading strategy for femto-cloud mobile network. Engineering Science and Technology, an International Journal, 19(1), 260-270.

[22] Boukerche, A., Guan, S., & Grande, R. E. D. (2019). Sustainable offloading in Mobile cloud computing: algorithmic design and implementation. ACM Computing Surveys (CSUR), 52(1), 1-37.

[23] Mohammed, M. A., & ŢĂPUŞ, N. (2017). A novel approach of reducing energy consumption by utilizing enthalpy in mobile cloud computing. Studies in Informatics and Control, 26(4), 425-434.

[24] Guo, S., Liu, J., Yang, Y., Xiao, B., & Li, Z. (2019). Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing. IEEE Transactions on Mobile Computing, 18(2), 319-333.

[25] Tang, C., Xiao, S., Wei, X., Hao, M., & Chen, W. (2018, January). Energy Efficient and Deadline Satisfied Task Scheduling in Mobile Cloud Computing. In 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 198-205).